2017

A Passive Brace for the Treatment of Scoliosis Utilizing Compliant Mechanisms

JB Ring
Bucknell University, jbr024@bucknell.edu

I, JB Ring, do grant permission for my thesis to be copied.

A PASSIVE BRACE FOR THE TREATMENT OF SCOLIOSIS UTILIZING COMPLIANT MECHANISMS

by

JB Ring

A Thesis

Presented to the Faculty of
Bucknell University
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Approved:

Charles J. Kim
Associate Professor of Mechanical Engineering,
Thesis Advisor

Jeffrey Evans
Chair, Department of Mechanical Engineering

Craig Beal
Assistant Professor of Mechanical Engineering,
Thesis Committee

Christine Buffinton
Associate Professor of Mechanical Engineering,
Thesis Committee

May 1, 2017

Acknowledgments

First, I would like to thank my advisor, Dr. Charles Kim, for inspiring me throughout my college career, beginning with statics class, and for guiding me through my graduate degree. I would like to express my gratitude to Dr. Christine Buffinton and Dr. Craig Beal for serving on my thesis committee and for their valuable feedback. All of Bucknell's faculty and staff who assisted with my research were instrumental to my success. These wonderful people include Dr. Katie Beiryla, Dr. Donna Ebenstein, Dr. Mala Sharma, Dr. Nate Sigel, and Dr. Indranil Brahma, along with Dan Johnson, Tim Baker, and Aaron Clark.

I would like to recognize the contributions of researchers at Laboratoire d'innovations CAO en génie orthopédique at the École Polytechnique de Montréal, Columbia University, and the Technical University of Delft. Specifically, I would like to thank Joep Nijssen from Delft. Working with Joep over the summer was an invaluable experience.

I would like to thank all of my friends that have accompanied me along this journey and my family who has supported me. I want to thank my fiancée, Alex, for pushing me to be the best. Lastly, I would like to thank Keeda for pulling me through the last two years.

Contents

List of Tables

List of Figures

Abstract

Adolescent idiopathic scoliosis is a deformity of the spine that affects 2-3% of the population and often requires treatment in the form of a brace. Most successful braces consist of a rigid plastic shell that can be uncomfortable and limit the patient's ability to perform activities of daily living. The greatest cause of treatment failure is patient unwillingness to wear the brace for the prescribed amount of time, up to 23 hours each day. Other flexible braces have been designed to overcome this obstacle, but they have a lower success rate and other drawbacks. It was proposed in this thesis that compliant mechanisms can provide the lateral stiffness required for correction combined with compliance in other directions since they naturally offer relative stiffness and compliance directions.

Throughout the process of designing the brace, multiple valuable contributions were generated for various fields of study. The development of the kinematic profile of the spine included determining the locations of the three primary spinal motions and specific axes of rotation for each motion. A corrective force paradigm was used for design rather than the standard displacement paradigm, therefore, requiring a complete understanding of the force profile applied by the brace, which is not found in literature. The force system was determined through an integration of the pressure applied to the torso by a brace. In order to design in 3-dimensions, the Building Block

Approach for compliant mechanism synthesis was expanded. This method was used to design the overall mechanism topology. Finally, an iteration of the brace was designed, produced, and tested. Overall, the tools necessary to design a compliant scoliosis brace were developed and can now be easily used to iterate through designs.

Chapter 1

Introduction

1.1 Scoliosis

Scoliosis, a musculoskeletal disease, presents as a three-dimensional deformity of the spine primarily characterized by curvature in the frontal plane [4, 19]. This thesis focuses on adolescent idiopathic scoliosis (AIS), which is the most common form of scoliosis and affects 2 to 3% of adolescents, approximately 10% of whom will require medical treatment [4, 20]. Girls are nearly three times more likely to have scoliosis than boys [21]. Scoliosis usually affects adolescents during their two growth periods from age five to eight and ten until the end of growth [22].

Scoliosis can lead to numerous health implications. Most notably, patients with scoliosis who require bracing or surgery can experience shortness of breath [22, 23]. In addition, scoliosis patients can suffer from heart problems and back pain [10, 22]. Struggling with self image is a nonphysical health implication, an emotional pain [22].

A scoliotic spine contains either an S or C curve (Figure 1.1a). The degree of scoliosis is characterized by the Cobb angle (Figure 1.1b), defined as the angle between the two most tilted vertebrae of a spine segment [24]. Cobb angles less than 25^o require biannual checkups but no treatment unless the angle increases. A Cobb angle greater than 40^o requires surgery. Cobb angles between 25^o and 40^o generally require bracing to prevent further progression [2].

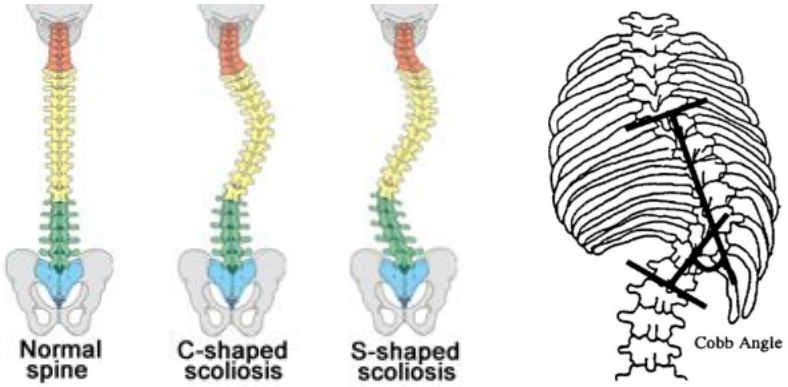


Figure 1.1: (a) Two standard shapes of the scoliotic spine and (b) a visual of the Cobb angle [2]

The most common cause of failure in brace treatment is a lack of patient compliance in wearing the brace [21]. Comfort and aesthetics are the main reasons that patients do not wear braces for the prescribed amount of time each day. Rigid braces limit range of motion, including flexion and rotation, which in turn limits the patient's ability to complete Activities of Daily Living (ADL). Braces are also bulky and cannot be easily hidden. Oversized clothes must be worn to hide a brace. This can damage self-confidence and have a psychological impact on the young patients who are usually going through puberty at the time of treatment [10].

1.2 Compliant Mechanisms

A compliant mechanism is a mechanisms that utilizes flexibility to accomplish something useful [17] and is composed of one or more compliant joints. Compliant joints can yield well-defined kinematic motion between bodies while providing relative force. They can be simply and inexpensively manufactured using processes such as casting, molding, machining, and various rapid prototyping methods [25]. Their use can reduce the number of components compared to a rigid body mechanism that uses springs [25] and are, as a consequence, lighter and simpler. They are capable of repeatable motion while undergoing minimal wear because they do not typically consist of contact joints. Fatigue is a dominant failure mode, which can be accounted for in design. There exists a strong correlation between fatigue life and maximum stress, so the life of a mechanism is easily predicted [14].

Although compliant mechanisms have many benefits, their disadvantages must also be considered. The size of a compliant mechanism limits its possible range of motion because a smaller mechanism reaches its yield strength at smaller deflections. In general, compliant mechanism fail at smaller loads than similarly sized rigid mechanisms [25]. In addition to their physical limitations, the coupled nature of a mechanism's force-displacement behavior makes their design and implementation a difficult challenge. Therefore, designing compliant mechanisms is rarely a straightforward process. The last difficulty to consider is designing for a mechanism's full range of motion. Most mechanisms are capable of nonlinear displacement and perform motions that cannot be described by a single point of reference. In addition to the variable kinematic characteristics of a mechanism, the stiffness of a mechanism fluctuates through its range of motion [14, 25]. All of these challenges must and will be addressed when designing the compliant mechanisms in this thesis.

1.3 Proposed Thesis Work

It is proposed that compliant mechanisms can be incorporated into a scoliosis brace to allow relatively unrestricted motion in the desired directions while maintaining stiffness in the corrective directions. Compliant mechanisms naturally experience relatively stiffness and compliant directions, which can be exploited for use in a flexible scoliosis brace.

The key contributions of this thesis are listed below and are reviewed in detail at the end of the thesis.

1. Spatial characterization of spinal motion including translational and rotational components.
   (a) Piece-wise description of where motions occur.
   (b) Specific axes of rotation of the motion of the spine.
2. Determining the magnitude of forces need to scoliosis correction.
3. Expansion of Building Block Approach for 3D use.
4. Use of the eigentwist to align a mechanism's rotation axis with a requirement.
5. New use for flexure mechanisms in the spatial problem of a scoliosis brace.

The first and second contributions were developed when characterizing the motion and force requirements for compliant mechanism design. The third contribution was used to design the topology of mechanisms for the brace. The fourth contribution allowed for the detailed dimensions of the mechanisms to be easily designed. The last contribution was the overall design of a new type of scoliosis brace utilizing compliant mechanisms.

Chapter 2

Literature Review & General Methodology

2.1 Scoliosis

To design the scoliosis brace, it was essential to first explore current braces and understand why they work. In designing the brace, an understanding of the kinematics of the spine was also needed, so previous studies of spinal motion were explored.

2.1.1 Bracing: State of the Art

Bracing is the most common treatment for AIS [4]. The goal of bracing is not necessarily to correct the curve, but to prevent further progression, though correction can occur [4, 10]. Braces are to be worn up to 23 hours a day [21]. The most common

braces are rigid, although concepts of flexible braces have recently been explored, and a few have been brought to market. Rigid braces include the Milwaukee, Boston, and Cheneau braces. Flexible braces include the SpineCor and the TriaC braces.

Rigid Braces

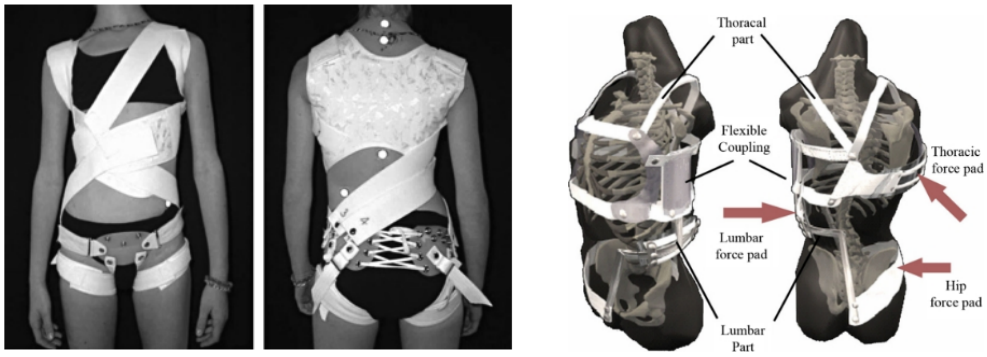
The Milwaukee brace (Figure 2.1a) was the first documented brace to prove effective with a 74% success rate [22]. It consists of a steel and leather pelvic base with rods that extend to the throat. However, the superstructure of this brace caused lower jaw and dental deformities [26]. The Boston brace (Figure 2.1b), currently most recommended for treatment, consists of a standard-sized polystyrene shell, tightened around the torso using straps, with interior foam padding to apply corrective forces and cut-outs to provide relief [2, 4]. The Boston brace has up to a 93% success rate [20]. The Cheneau brace (Figure 2.1c), which has many variations, also has a rigid plastic shell, but is customized to each individual patient using a hyper corrected positive plaster model [27].

(a) **(b)** **(c)**

Figure 2.1: (a) Popular brace designs include Milwaukee [3] , (b) Boston [4], and (c) Cheneua [5]

Flexible Braces

Flexible braces are designed to address the drawbacks of rigid braces. The SpineCor brace (Figure 2.2a) is a flexible brace consisting of elastic bands, a pelvic base, and crotch and thigh bands [28]. This brace allows the patient a much greater range of motion compared to rigid braces, but has a lower success rate according to some sources [28, 29]. Guo et al. found that 5 out of 7 patients who encountered progression of the spinal deformation with the SpineCor brace had no further progression after switching to a rigid brace [29]. The patient also requires assistance to put on the brace. The TriaC brace (Figure 2.2b) is designed with regard to the three C's (Comfort, Control, and Cosmetics) proposed by Nijenbanning. Comfort refers to the patient's ability to perform ADL. Control refers to the brace's ability to apply correction forces of the right directions and magnitudes. Cosmetics refers to the appearance of the brace itself, along with how the patients perceive themselves in the brace [10]. This brace consists of straps, force pads, and a flexible coupling. In a study of 63 patients, it was reported that the TriaC brace had a 76% success rate [7]. In general, flexible braces are less effective than rigid braces in preventing scoliotic progression.



(a) **(b)**

Figure 2.2: (a) Front and back views of SpineCor brace [6] and (b) TriaC Brace [7]

Brace Conclusions

It is concluded from this review that rigid braces have higher rates of successful treatment because they are capable of greater control, and that the compliance of flexible braces inherently causes loss of control resulting in increased comfort but lower success rates. This thesis proposes that there must be some middle ground to create a controlled comfort, and that the solution is compliant mechanisms. Compared to the straps of the SpineCor brace, it is proposed that compliant mechanisms can provide more control while allowing motion. This controlled comfort will be developed throughout the process of designing the brace.

2.1.2 Characterization of Spinal Motion

Many studies, including [1, 9, 30, 31], have been performed to characterize the motion of the spine. Each of these studies characterizes the motion of the spine as comprised of three primary motions: flexion, lateral bending, and twist. Flexion is bending in the sagittal plane, lateral bending is in the coronal place, and twist is rotation about the vertical axis. Refer to Figure 2.3 for the planes of the body. The studies mentioned present a list of ADL along with the percentage of each primary motion needed to complete them (Table 2.1) which can be used to determine some general requirements for the brace.

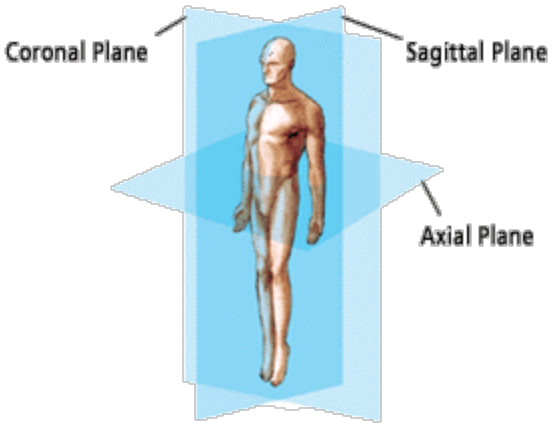


Figure 2.3: Planes of the body used to describe motion [8].

Table 2.1: List of Activities of Daily Living and percent of full range of each primary motion necessary to complete them [1].

	Average Percentage of Full Active ROM		
ADL	Flexion/Extension	Lateral Bending	Axial Rotation
Stand to sit	37	20	12
Backing Car	10	16	18
Reading	4	6	6
Feeding	5	8	9
Socks	22	19	14
Shoes	20	20	16
Sit to stand	39	14	10
Washing Hands	12	15	12
Make-up	7	11	8
Squatting	52	31	18
Bending	59	29	18
Walking	11	19	19
Up stairs	13	22	20
Down stairs	11	21	18
Maximum	59	31	20
Average	22	18	14

To gain the desired control needed to provide correction, it is necessary to know where specifically in the spine each of the three primary motions predominately occurs. Three studies measured relative motion between the L1 and S1 vertebrae using

a sensor on each vertebra [1, 30, 31]. One used motion capture with markers at different vertebrae, but only used the S2 and C7 vertebrae to calculate the motion angle as explained in Figure 2.4 [9]. These studies only identified the planar motion of the spine, assuming that flexion only needed to be considered in the sagittal plane, lateral bending only in the coronal plane, and twist only in the axial plane. However, to design a brace that properly follows the motion of the torso, a spatial characterization describing the 3D rotation and translation of motion was required. In addition, the studies did not describe any smaller subregions of the lumbar or thoracic spine. This type of piecewise description was desired because it allowed for a better understanding of which regions of the spine contributed most to each of the three primary motions and helped determine where mechanisms could be placed. The study performed to gain this knowledge is seen in Section 3.2.

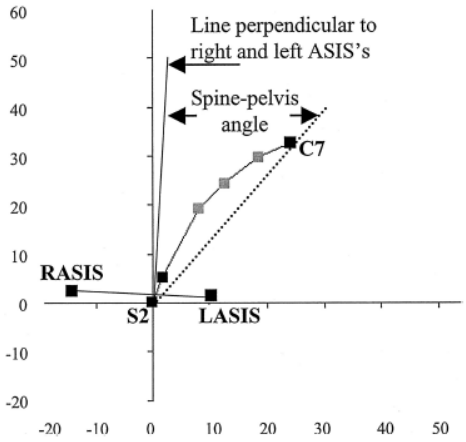


Figure 2.4: Motion study where markers were placed on several vertebrae and the patient performed lateral bending. The motion of the spine was characterized in a plane as the angle between RASIS/LASIS line and S2/C7 line [9]. This characterization was a point-to-point analysis and did not consider the rotation and translation that fully define the motion.

2.1.3 Correction Principle and Forces

In addition to the kinematic requirements, force requirements are needed for compliant mechanism design. However, the standard correction principle for bracing, as used by the Boston Brace, focuses on displacement correction rather than force correction. This standard principle is to displace the spine into a corrected state in order to prevent progression. For example, Desbiens-Blais et al. suggests about 30mm of displacement to correct an initial curve of Cobb angle 30° [24]. Since the braces are rigid, the patient can move their torso away from the padding and therefore the correctional force. This is known as the two-fold principle of correction [32], with the primary goal of displacing the spine and the secondary goal of allowing self-correction. Nijenbanning believed this principle to be contradicting in that during self-correction, the brace serves no function. Therefore, a new bracing principle was established in which "the brace forces must be able to follow the main body motion of the patient" [10]. This principle was used as the basis for the design of the TriaC Brace. Neither of the principles can be directly applied to the compliant brace because the force and motion of compliant mechanisms are highly correlated. A new paradigm explaining this force-motion continuum is presented in Section 3.3.1.

Literature regarding the biomechanics of scoliosis bracing contains a minimal amount of information pertaining to the forces needed for correction because most braces use a displacement criterion for correction.. Nijenbanning provides an in-depth review of the available literature of bracing forces, finding that 3-point (for C-curve) and 4-point (for S-curve) force systems are common among rigid scoliosis braces [10]. He concluded that the 4-point force system in Figure 2.5 would be used for the TriaC brace. This force system is confirmed by the *Reference Manual for the Boston Scoliosis Brace* which states that the Boston brace is designed to apply forces at and

below the apices of the spinal curve [32]. Physically, this is sensible because pushing up decompresses the spine and pushing inward straightens the spine.

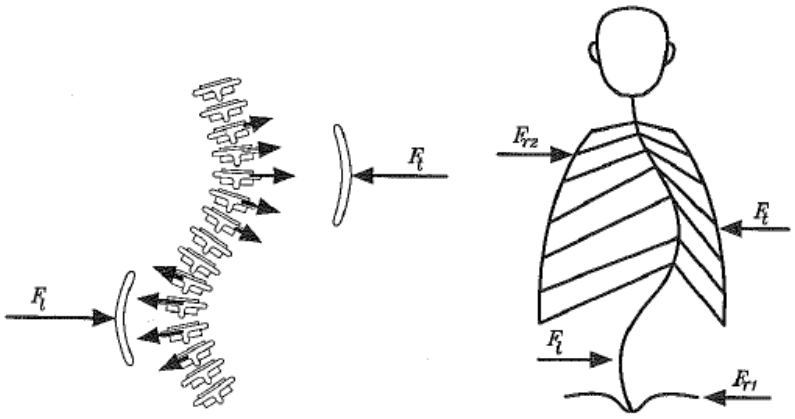


Figure 2.5: Proposed correction system for TriaC brace [10].

The necessary magnitude of these forces is not well-known. A magnitude of 20-40 Newtons of lateral force was suggested by Nijenbanning [10], but this is the only source that estimated the magnitude of correction forces. To determine the actual force magnitudes necessary for correction, a method involving integrating a pressure map of the pressure applied to the torso by the brace was devised. The pressure was derived from a Finite Element (FE) model of the torso with the applied brace [33]. This analysis, seen in Section 3.3, also resulted in locations and directions of forces.

2.2 Compliant Mechanisms

There are many types of compliant mechanisms: shell, flexure, lamina emergent, origami, etc. Flexure mechanisms are the primary type of mechanism used in this thesis because they are the most researched, and many flexure mechanisms are well known and understood. They are comprised of an arrangement of flexure elements such as beams. Flexures prefer a certain motion dependent on the orientations of the

beams to one another. Flexure mechanisms were used for this thesis because they are the most studied and well-understood. Therefore, their behavior is predictable when using them in new applications such as the scoliosis brace.

2.2.1 Example Flexures

The simplest element is a single-beam (Figure 2.6a), which is equivalent to a cantilever. Two-beam flexures, designed for translational motion, can be compared to a four bar mechanism (Figure 2.6b). When both beams in a two-beam flexure are the same length, they produce parallel motion. The two-beam flexure, however, suffers from parasitic motion normal to the applied force and desired motion direction [14]. When a force is applied as seen in Figure 2.6b, the resulting motion is translation down and to the left.

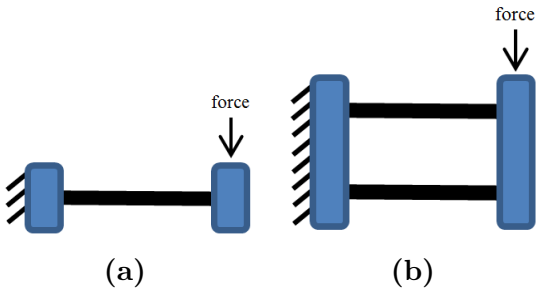


Figure 2.6: (a) Single-beam flexure and (b) two beam flexure.

Another embodiment of a translational joint is the Compliant Translational (CT) joint (Figure 2.7) [34]. The CT Joint leverages an arrangement of parallelograms to eliminate parasitic bending and transverse displacement.

The cartwheel hinge and the cross-strip-pivot, displayed in Figure 2.8, are designed for revolute motion. The cartwheel hinge maintains a single axis of rotation when

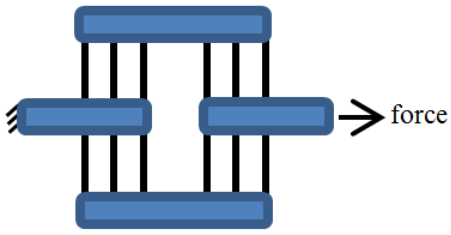
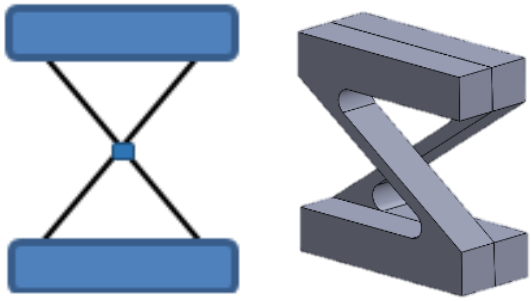


Figure 2.7: CT Joint

deflected; however, the axis of rotation of the cross-strip pivot changes with deflection. The cross-strip pivot's design results in a lower stiffness [25].



(a) **(b)**

Figure 2.8: (a) Cartwheel hinge and (b) Cross-strip pivot.

There are a great number of compliant flexure joints. Trease et al. [34] provide a review and comparison of many.

2.2.2 Synthesis Methods

In compliant mechanism design, synthesis methods are used to determine the topology and dimensions of mechanisms. The main synthesis methods used to design compliant mechanisms can be seen in Table 2.2. All are useful for certain design problems, but none suit the needs of the scoliosis brace design problem. A brief explanation of each

is listed below.

Table 2.2: Standard compliant mechanism synthesis methods and their general characteristics.

Synthesis Method	2D/3D	Problem Statement Form	Mechanism Type
Rigid Body Replacement	2D	Analogous rigid mechanism design	Flexure
FACT	3D	Freedom and constraint directions	Flexure
Topology Opt.	2D/3D	Design space, Desired reaction motion to applied force	Flexure
Building-Block	2D	Compliance and stiffness direction represented by an ellipse	All

In *Rigid Body Replacement*, a mechanism is designed using rigid links and standard joints. Then attempts are made to replace the standard joints with compliant joints. *Pseudo-Rigid Body Modeling (PRBM)* is the theoretical tool used for this method. It allows the designer to model the kinematics and kinetics of a mechanism by assuming a rigid linkage with springs at the joints [14]. For a simple compliant beam (Figure 2.9), a pin joint can be placed at some known distance from the end which is used to approximate the complaint motion. This known distance is the characteristic radius. A spring of a certain stiffness, calculated with respect to the specific beam, can be applied at the joint. This combination of joint position and spring stiffness provides a reliable model for small deflections. This same theory can be applied to various joint types and boundary constraints.

Freedom and Constraint Topology (FACT) uses a library of geometric shapes (Figure 2.10) that account for the freedom and constraints of a problem and allow the designer to visualize where compliant elements could be placed to achieve the desired motion. The designer categorizes their problem as one of the freedom space in the

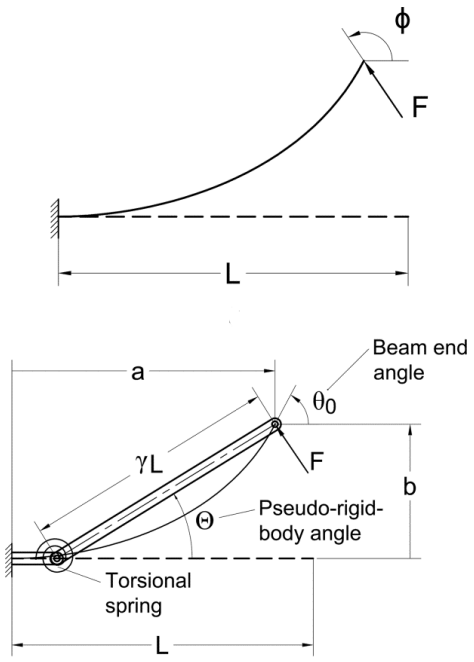


Figure 2.9: Example of psuedo-rigid body model (bottom) for simple compliant beam (top) [11].

library and uses its complimentary constraint space for design [17].

In *Topology Optimization*, the designer provides a workspace of possible material connections and allows the computer to optimize the distribution of material and voids within the workspace. The designer also provides a desired input force and output displacement [17]. The most common example is the force inverter seen in Figure 2.11.

The *Building-Block Approach* involves characterizing the problem in the same way as a library of mechanisms and comparing the characterization to the library to find possible solutions. The specific characterization method involves an eigenvalue decomposition of the compliance matrix to define the primary compliance and stiffness directions of a mechanism. These are mapped to an ellipse to create a visual for comparison. This idea of the compliance ellipse is further explored in Section 4.2.1

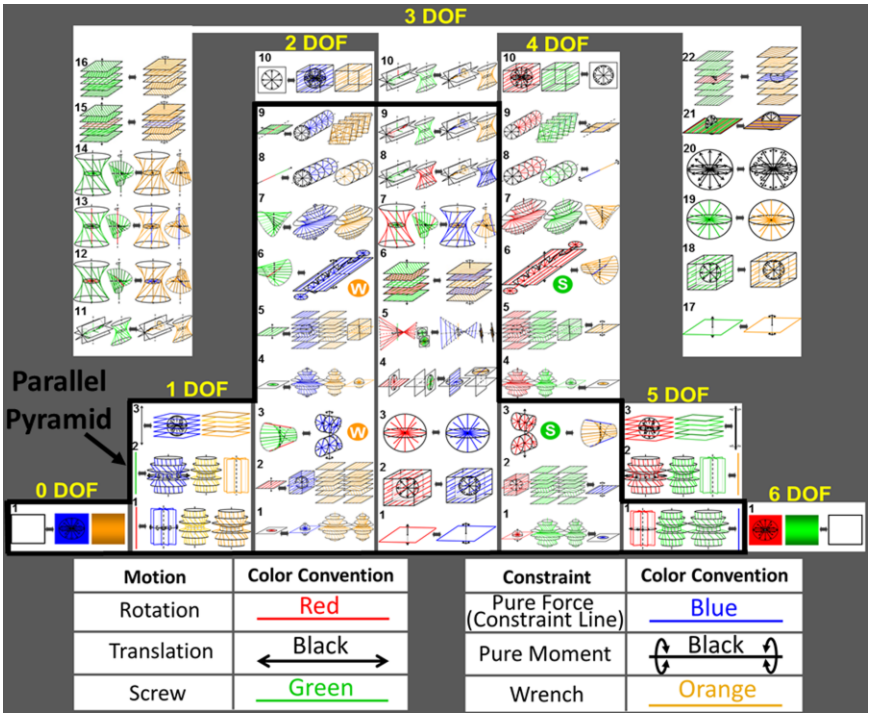


Figure 2.10: Library geometric shapes used in FACT [12].

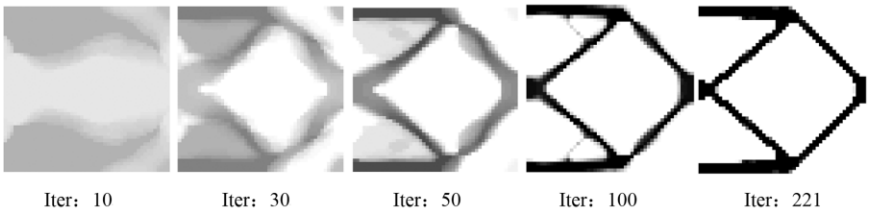
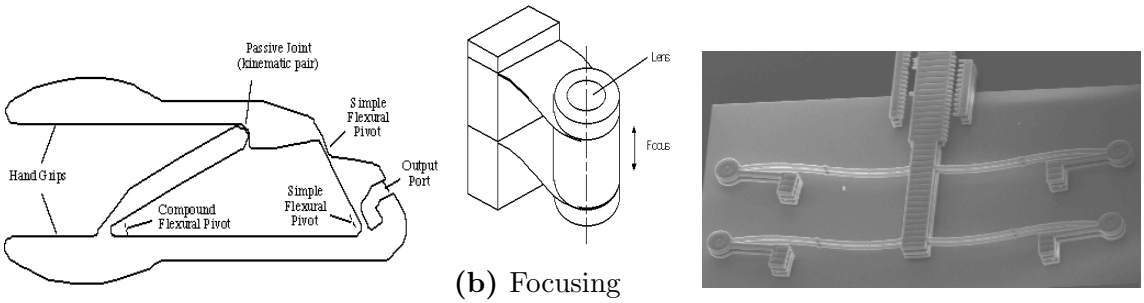


Figure 2.11: Topology optimization of force inverter where a force applied on the left side pushing to the right causes horizontal compression of the mechanism [13].

for use in designing the scoliosis brace.

Considering the four synthesis methods, it was decided to expand the Building Block Approach for use in 3D. Other researchers are developing tools to use Rigid Body Replacement in 3D, but this methodology was not used because there is not already a rigid mechanism solution that could be replaced, as is necessary for the process. FACT does offer full 3D synthesis, but is limited to flexure mechanisms. Topology Optimization is computationally intensive and allows the designer to pro-

vide limited input. In 3D, it would take a very large amount of time. To produce contributions to the field of compliant mechanism for this thesis, it was decided that the Building Block Approach would be furthered because it has the adaptability to be used for all types of mechanisms.



(a) Pliers **(b)** Focusing Lens **(c)** MEMS Switch

Figure 2.12: Example uses of compliant mechanisms [14].

2.3 General Methodology

The process for designing mechanisms for the scoliosis brace is a fine balance between design for motion (kinematics) and design for force (kinetics) This is natural with compliant mechanism design because force and displacement are directly correlated through the mechanism's stiffness. This consideration is emphasized by the scoliosis brace problem because both the motion and force requirements are essential to success. The brace must remain appropriately stiff to correct scoliosis, but must also allow motion to improve patient comfort.

The general procedure used in the design of this brace was:

1. Determine kinematics and kinetic requirements for scoliosis bracing.
2. Define the topology of the brace pieces.

3. Define the characteristic requirement ellipsoids based on the brace piece topology.
4. Compare characteristic ellipsoids to library of mechanism ellipsoids to choose possible mechanisms.
5. Align primary eigentwist of mechanism with kinematic requirement screw to determine the general mechanism dimensions.
6. Determine the optimal beam thickness as determined by the kinematic and kinetic magnitude requirements.

The first step of designing the brace, defining the kinematic and kinetic requirements, is explained in Chapter 3. Both types of requirements were characterized by their magnitudes, location, and directions. The kinematic requirements were determined through a motion study of the torso motion and were composed of axes of rotation located in space and the magnitudes of rotation. The forces that the brace imparts on the torso were determined through an integration of the pressure applied to the torso by the brace. These forces were used as the kinetic requirements.

After the requirements were known, the design of the brace was performed starting with the topology of the brace shell and the topology of the mechanisms (Section 4.2). Instead of creating a completely new brace for a patient, a brace from a successful case study was used. This ensured that the geometry of the brace and the necessary force system was realistic. Material was removed from the brace to create three separate brace pieces to be connected by mechanisms. The location of removed material was determined relative to the locations of the force and motion requirements. The force system on each brace pieces was known and therefore the relative forces between brace pieces could be calculated, resulting in the required mechanism forces. These forces were later used in determining the ideal geometry of the mechanisms.

The topology design of the brace system provided one major contribution of this thesis. The Building Block Approach was expanded for 3D usage. The brace requirements were characterized into the same visualization as the mechanism library for easy comparison, and mechanisms were chosen to be implemented. The general design for the brace was then complete.

The dimensional design (Section 4.3) of the brace began with aligning the mechanism's axis of rotation with the requirement axis of rotation. All geometric parameters of the mechanisms, except thicknesses, affected the location of the rotation axis. A tool utilizing the eigentwist was developed and used to align the mechanism's rotation with the requirement. Lastly, the ideal beam thickness was determined such that the mechanisms performed the required motion while applying the required force system.

The order of the steps in this process were important because each step informed the design of the next step in some way. The brace pieces informed the locations and geometric limitations of the mechanisms. The topology design, as a whole, informed the dimensional design. Lastly, the ideal thicknesses of beams were affected by other geometric parameters of the mechanisms.

The requirements that were needed in each part of the design process are noted in Table 2.3. It should be noted that the magnitudes of the motion and forces were not utilized until the last step of the process, determining beam thickness.

Table 2.3: Describes the requirements used in each part of the design process.

		Location / Direction		Magnitude	
	Design Steps	Force	Motion	Force	Motion
Topology Design	Brace Pieces	●	●		
	Characteristic Ellipsoid	●	●		
Detail Design	General Mechanism Dimensions		●		
	Beam Thickness			●	●

Chapter 3

Requirement Generation

Motion and force requirements are developed in this chapter. Some general brace requirements are introduced first. Then the process of determining the required magnitudes, directions, and locations of motions is detailed in Section 3.2. Motion capture was taken of a subject performing the three primary motions, and screw theory was used to develop a spatial description of the motions. The method of acquiring force requirements for design is described in Section 3.3. The method relied on integrating a mapping of the pressures applied to the torso by a brace.

3.1 General Requirements

The brace must be relatively comfortable since it must be worn for most of the day. The main controllable factor in comfort is minimizing edges that dig into the patient. In addition, the patient must be able to put the brace on with minimal assistance.

Aesthetic considerations should also be considered. The average patient will not be content with a brace that is too unappealing. A brace that is large or considered an eyesore is unlikely to be worn for the prescribed amount of time, even if it is more comfortable and allows greater ranges of motion. This is the primary reason for failed treatment in rigid braces, so it should be also avoided in the flexible brace.

3.2 Motion Requirements

The motion of the spine has been heavily studied [1, 9, 30, 31], but this research has been with respect to the planar motion of a large section of the spine, such as the lumber or thoracic regions. Literature provided no spatial description of the spine describing the translational and rotational components that fully define the motion, nor did it describe where in the spine each of the primary motions predominately occurs.

Literature was able to provide insight on the magnitude of motion required to complete certain functions. The information provided in the literature assisted in developing the magnitude requirements for motion by presenting how much of the full range of motion was required to complete activities of daily living (ADL) [1]. A motion study was performed to determine the information missing from literature and is described in this section.

3.2.1 Activities Of Daily Living - Range of Motion

With knowledge of the maximum and average percentages of motion needed to perform ADL noted in Table 3.1, the goal percentages were decided to be 40%, 30%, and 20% for flexion, lateral bending, and twist, respectively. Bending requires the most flexion at 59% of full ROM, with sitting requiring the second most, 39%. The value of 40% was chosen for flexion because sitting is a necessity, but full bending is not. 30% and 20% of lateral bending and twist, respectively, are sufficient to complete all ADL.

Table 3.1: Truncated list of activities of daily living and percent of full range of each primary motion necessary to complete them, from Section 2.1.2 [1].

	Average Percentage of Full Active ROM		
ADL	Flexion/Extension	Lateral Bending	Axial Rotation
Stand to sit	37	20	12
Sit to stand	39	14	10
Squatting	52	31	18
Bending	59	29	18
Maximum	59	31	20
Average	22	18	14

3.2.2 Motion Capture & Screw Analysis

Although literature provided valuable insight into the motion of the spine, a piecewise description of the spatial motion was required. Ideally, the relative motion between each consecutive vertebra needed to be known. The desired piece-wise description provided knowledge of which regions of the spine are responsible for which primary motions. For instance, it could be found that flexion only occurs in the lower lumber region of the spine. Knowing where the motion specifically occurs will allow for placement of mechanisms that better follow the natural spinal motion. It was

determined that capturing the motion (displacement and rotation) of a number of vertebrae through primary motions provided the data needed. From the analysis, the following resulted:

- Specific full range of motion
- Locations and directions of axes of motion for flexion, lateral bending, and twist
- Piecewise characterizations of where motions occur in the spine

Experimental data was acquired at 100 Hz (frames/sec) from a human subject using a VICON motion tracking system with 8 T10 series cameras. The motion capture data was analyzed for primary motion behavior using homogeneous transformations and screw theory. Screw theory provides an intuitive representation of the captured motions. Screw theory states that any rigid body motion can be represented by a rotation and a translation in the direction of the axis of rotation [35] as seen in Figure 3.9. The parameters used to define the screw are:

$\vec{\omega}$	direction of screw axis
$\vec{q}$	point in space that locates the screw axis
h	pitch of the screw
θ	rotation about the screw

Screw theory made it possible to determine whether a rigid body motion is primarily translation, rotation, or a combination of the two. As seen in Figure 3.2, five sets of three markers were placed at the S1, L4, L1, T8, and T4 vertebrae. Each set of three markers was assumed to represent a rigid body that could be analyzed using screw theory. The test subject, not shown in the figure, was a 20 year-old female of approximately 5 feet and 6 inches who had successfully been treated for scoliosis through use of a brace. For each data set, the subject was instructed to perform one of the three primary motions three times.

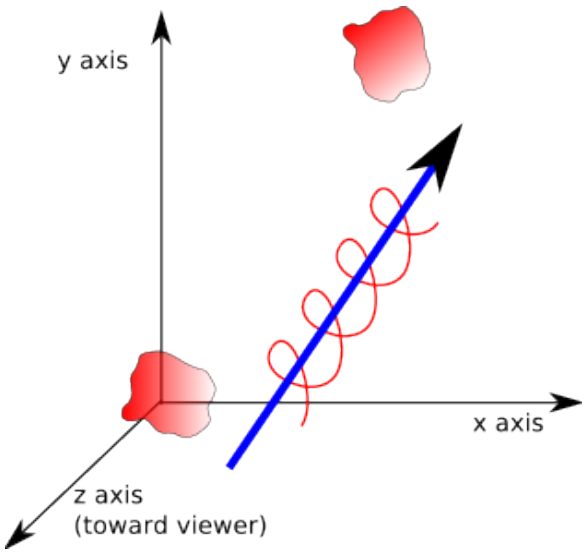


Figure 3.1: Visualization of screw theory [15].

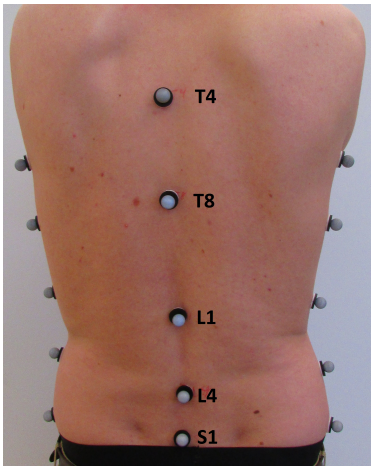
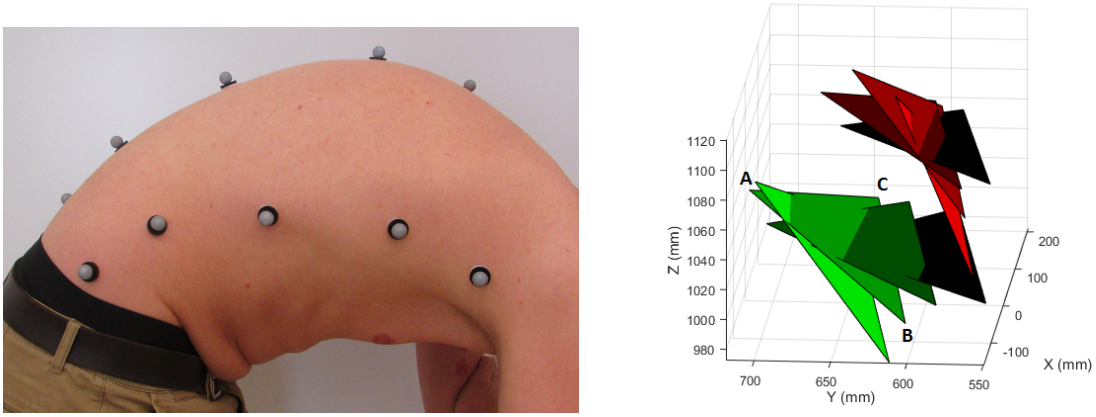


Figure 3.2: Example of marker placement on the torso. Note that this was not the subject used in design of the brace.

Screw Theory Analysis

For simplicity and to explain the this analysis, only the lower two sets (S1 and L4) in the flexion motion were considered. A sample of four time steps of raw motion data throughout the flexion is shown in Figure 3.3b.

The goal of the analysis was to characterize the primary motion of the L4 set in



(a) Five Sets of Markers **(b)** Sagittal Bending

Figure 3.3: (a) Subject performing flexion, (b) Visualization of raw motion data in the global frame, S1 set (green), L4 set (red). The darker shades represent the initial position, and lighter colors represent motion through flexion.

the reference frame of the S1 set. To simplify that analysis, the bodies in each time frame were transformed so that the S1 set was always in the same position. Euler angles were used for this conversion. For consistency, the back marker, right marker, and left marker are referred to as A, B, and C, respectively (Figure 3.3b). First, the vector between vertex A and the origin of the global frame was calculated. Each point of both the S1 and L4 sets was transformed along this vector so that the vertex A was located at the origin. The i subscript implies that this process is looped through each time frame, one at a time.

$$A_i' = [0, 0, 0] = A_i - A_i \tag{3.1}$$

$$B_i' = B_i - A_i \tag{3.2}$$

$$C_i' = C_i - A_i. \tag{3.3}$$

Next, each AB leg of the triangle was rotated about the y-axis into the xy-plane. The angle between the x- and z-components of B was found using the four quadrant

inverse tangent. This angle was used as an input for the y-axis rotation matrix. Each three-dimensional point (A_i', B_i', and C_i' for both bodies), represented by L, of the two sets was transformed using

$$L' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_y(\theta) * L = \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{3.4}$$

The same leg of the triangle was then rotated about the z-axis to be aligned with the x-axis. The necessary angle of rotation between the x and y components of B was again calculated using the four quadrant inverse tangent, and

$$L'' = \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = R_z(\theta) * L' = \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{3.5}$$

was used to transform the points.

Finally, the other leg, AC, was rotated about the x-axis to be in the xy-plane. The same method used

$$L''' = \begin{bmatrix} x''' \\ y''' \\ z''' \end{bmatrix} = R_x(\theta) * L' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \tag{3.6}$$

for this final rotation. These four steps are visually represented in Figure 3.4.

Throughout this series of transformations, the relative positions of the S1 and L4 sets stayed consistent. Figure 3.5 displays the L4 set moving in the frame of the

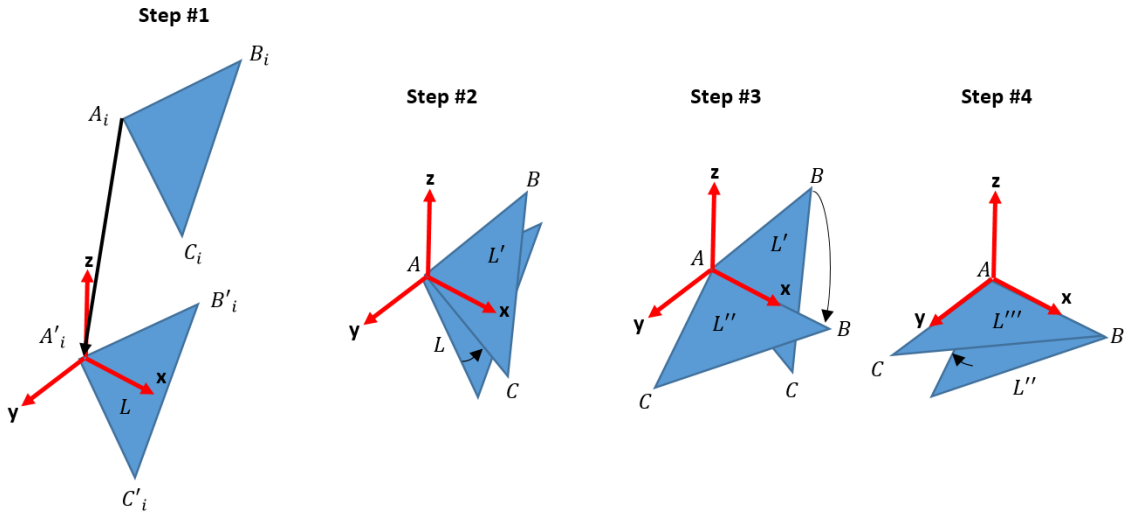


Figure 3.4: The four steps used to translate and rotate the L4 marker set into the reference frame of the S1 set of markers.

stationary S1 set.

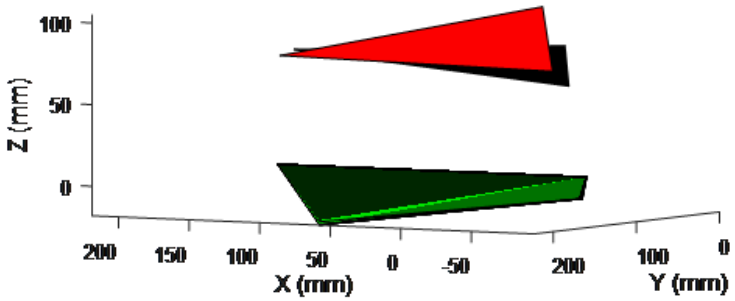


Figure 3.5: The green quadrilateral at the bottom represents the stationary S1 set. The starting and final positions of the L4 set are represented by the red triangles towards the top. The green quadrilaterals seem to not be coincident due to small numeral inconsistencies.

The sets were initially modeled as rigid. The motion data had small error due to

deformation of the mid set that caused the exact dimensions of the L4 set to vary. To overcome this error, the dimensions of the set in the first time frame were imposed on the sets in each of the other time frames. Points A, B, and C were used as previously defined to describe this conversion. The length of AB and AC were calculated, along with the angle BAC for the first time frame of data.

To convert the sets to rigid bodies in each time frame, the following procedure was performed. The unit vectors for $\vec{AB}$ and $\vec{AC}$ were calculated as follows:

$$\hat{AB} = \frac{\vec{AB}}{\left|\vec{AB}\right|} \tag{3.7}$$

and

$$\hat{AC} = \frac{\vec{AC}}{\left|\vec{AC}\right|}. \tag{3.8}$$

The cross product of these vectors was used to determine the orthogonal vector

$$\vec{k} = \hat{AB} \times \hat{AC}. \tag{3.9}$$

$\vec{AB}$ from the current time frame and the BAC angle from the first time frame were input into Rodrigues Formula,

$$\hat{AC'} = \hat{AB}cos\theta_{BAC} + (\vec{k} \times \hat{AB})sin\theta_{BAC} + \vec{k}(\vec{k} \cdot \hat{AB})(1 - cos\theta_{BAC}), \tag{3.10}$$

in order to calculate the new $\vec{AC}$ unit vector direction. Lastly, $\hat{AB}$ and $\hat{AC'}$ were multiplied by their respective magnitudes as calculated from the first time frame. Performing this process in each time frame resulted in the sets of each time frame being the exact same size.

Screw theory was now used to describe the motion of the L4 set relative to the S1 set. Screws for the movement of the L4 set from the first timeframe to each other timeframe in the dataset were found in the spatial frame of the S1 set using the following procedure.

The points A, B, and C were used in this procedure and are defined as shown in Figure 3.6. The points with subscript 0 represent the first time frame of the set, while the subscript i represents any one time frame in the motion.

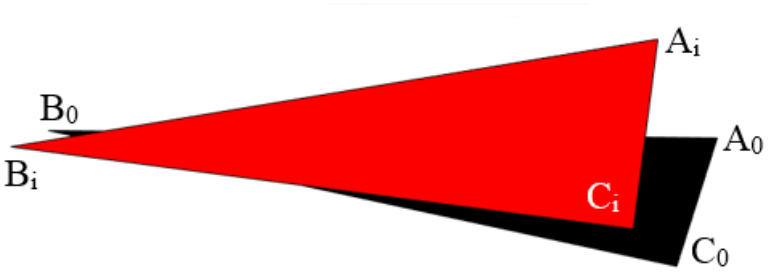


Figure 3.6: Labeled visualization of initial set position and time frame position.

The screw is defined by a direction vector, a point on that vector, a screw pitch, and an angle of rotation. First, the axis of rotation was determined. A vector $\vec{d}$ was defined as

$$\vec{d} = A_i - A_o. \tag{3.11}$$

The vector $\vec{d}$ was subtracted from A_i, B_i, and C_i, translating the current body so that A_i is coincident with A_0. The screw direction vector is defined by the transformation of B and C and must be orthogonal to the vectors,

$$(B_i - \vec{d}) - B_0 \tag{3.12}$$

and

$$(C_i - \vec{d}) - C_0 \tag{3.13}$$

and therefore the cross-product was used to calculate the direction unit vector of the screw,

$$\frac{(B_i - \vec{(}d) - B_0) \times (C_i - \vec{d} - C_0)}{\left\|(B_i - \vec{(}d) - B_0) \times (C_i - \vec{d} - C_0)\right\|}. \tag{3.14}$$

Next, the angle of rotation was determined by considering movement of a point about the rotation axis (Figure 3.7). Considering the point C as it translates from C_0 to $C_i - \vec{d}$, the angle was defined by the perpendicular vector from the rotation axis to the point C_0 and the perpendicular vector from the rotation axis to the point C_0 to $C_i - \vec{d}$. These values were defined as

$$\vec{u'} = \vec{u} - \vec{\omega} \cdot \vec{u} \tag{3.15}$$

and

$$\vec{v'} = \vec{v} - \vec{\omega} \cdot \vec{v}. \tag{3.16}$$

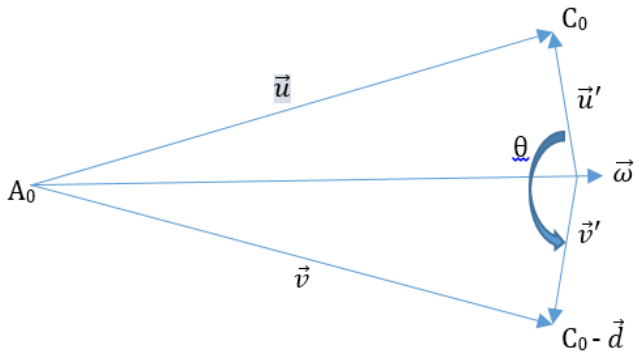


Figure 3.7: Representation of rotation about screw.

The angle of rotation was found using Eq. 3.19, which was derived through Eqs. 3.17 and 3.18.

$$sin\theta = \frac{\left\|\vec{u'} \times \vec{v'}\right\|}{\left\|\vec{u'}\right\| \left\|\vec{v'}\right\|} \tag{3.17}$$

$$cos\theta = \frac{\vec{u'} \cdot \vec{v'}}{\left\|\vec{u'}\right\| \left\|\vec{v'}\right\|} \tag{3.18}$$

$$tan\theta = \frac{sin\theta}{cos\theta} = \frac{\left\|\vec{u'} \times \vec{v'}\right\|}{\vec{u'} \cdot \vec{v'}} \tag{3.19}$$

Since $\vec{u'} \times \vec{v'}$ is parallel to $\vec{\omega}$ and the magnitude of $\vec{\omega}$ is unity, it can be seen that

$$\left\|\vec{u'} \times \vec{v'}\right\| = (\vec{u'} \times \vec{v'}) \cdot \vec{\omega}. \tag{3.20}$$

The angle was calculated using Eq. 3.19 and the four quadrant inverse tangent

$$\theta = tan^{-1} \frac{(\vec{u'} \times \vec{v'}) \cdot \vec{\omega}}{\vec{u'} \cdot \vec{v'}}. \tag{3.21}$$

The pitch of the screw was determined next. Using a different form of the Rodrigues formula,

$$R = e^{\hat{w}\theta} = I + \hat{w}sin\theta + \hat{w}^2(1 - cos\theta)) \tag{3.22}$$

we have

$$T = \begin{bmatrix} R & \vec{d} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} e^{\hat{w}\theta} & (I - e^{\hat{w}\theta}\vec{q}) + h\theta(\vec{\omega}) \\ 0 & 1 \end{bmatrix} \tag{3.23}$$

where R is a rotation matrix and $\vec{d}$ is the displacement vector. The translation component was separated into the components parallel to $\vec{\omega}$ and perpendicular to $\vec{\omega}$. The following relationship between the parallel component and the project of $\vec{d}$ into $\vec{\omega}$ was used to determine the pitch of the screw:

$$\vec{d} \cdot \vec{\omega} = h\theta \tag{3.24}$$

where all values are already know other than the pitch, h.

The last component of the screw found was the point that locates the screw, $\vec{q}$. There are an infinite number of points that could be used to define the screw. $\vec{q}$ is defined as the point on the screw closest to the zero of the global frame. Refer to Figure 3.8 for assistance in understanding the following process.

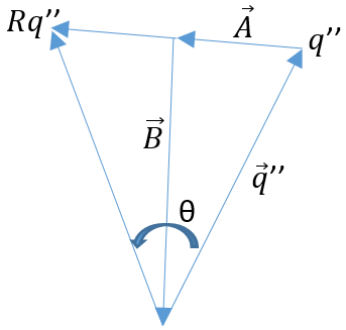


Figure 3.8: Representation of calculation for screw location point.

The vector, $\vec{q''}$, which can be defined as,

$$\vec{q''} = -(\vec{A} + \vec{B}) \tag{3.25}$$

was sought to locate the screw in space. The vectors $\vec{A}$ was defined as

$$\vec{A} = -\frac{\mathbf{R}\vec{q} - \vec{q}}{2} = \frac{(\mathbf{I} - \mathbf{R})\vec{q}}{2}. \tag{3.26}$$

Through trigonometric relations,

$$tan\left(\frac{\theta}{2}\right) = \frac{\left\|\vec{A}\right\|}{\left\|\vec{B}\right\|} \Rightarrow \left\|\vec{B}\right\| = \frac{\left\|\vec{A}\right\|}{tan\frac{\theta}{2}} = \frac{\|(\mathbf{I} - \mathbf{R})\vec{q}\|}{2tan\frac{\theta}{2}} \tag{3.27}$$

the magnitude of $\vec{B}$ was determined. The direction of $\vec{B}$ was defined as

$$-\vec{\omega} \times (\mathbf{I} - \mathbf{R})\vec{q} \tag{3.28}$$

such that

$$\vec{B} = \frac{\|(\mathbf{I}-\mathbf{R})\vec{q}\|}{2tan\frac{\theta}{2}}\left(\frac{-\vec{\omega}\times(\mathbf{I}-\mathbf{R})\vec{q}}{\|\vec{\omega}\times(\mathbf{I}-\mathbf{R})\vec{q}\|}\right). \tag{3.29}$$

Note that

$$\|\vec{\omega}\times(I-\mathbf{R})\vec{q}\| = \|(I-\mathbf{R})\vec{q}\| \tag{3.30}$$

because the magnitude of $\vec{\omega}$ is one and $\vec{\omega}$ is perpendicular to $(\mathbf{I}-\mathbf{R})\vec{q}$. Hence,

$$\vec{B} = \frac{-\vec{\omega}\times(\mathbf{I}-\mathbf{R})\vec{q}}{2tan\frac{\theta}{2}}. \tag{3.31}$$

Therefore,

$$\vec{q''} = \frac{(\mathbf{I}-\mathbf{R})\vec{q}}{2} + \frac{-\vec{\omega}\times(\mathbf{I}-\mathbf{R})\vec{q}}{2tan\frac{\theta}{2}} = \left(\mathbf{I}+\frac{\vec{\omega}\times}{2tan\frac{\theta}{2}}\right)\frac{(\mathbf{I}-\mathbf{R})\vec{q}}{2} \tag{3.32}$$

where $\vec{\omega}$ is expressed as a skew symmetric matrix. To simplify the equation, recall that

$$(\mathbf{I}-\mathbf{R})\vec{q} = \vec{d} - h\theta\vec{\omega} \tag{3.33}$$

so that

$$\vec{q''} = \left(\mathbf{I}+\frac{\vec{\omega}\times}{2tan\frac{\theta}{2}}\right)\frac{\vec{d}-h\theta\vec{\omega}}{2}. \tag{3.34}$$

An example of the resulting screws from this procedure are represented by the multicolored vectors in Figure 3.9.

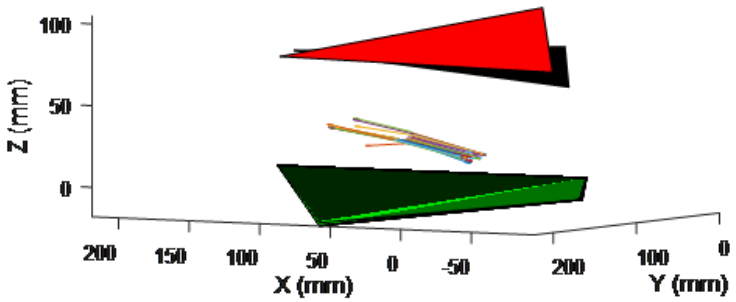


Figure 3.9: Labeled visualization of screws with motion of L4 set.

Post-Processing Screw Theory Output

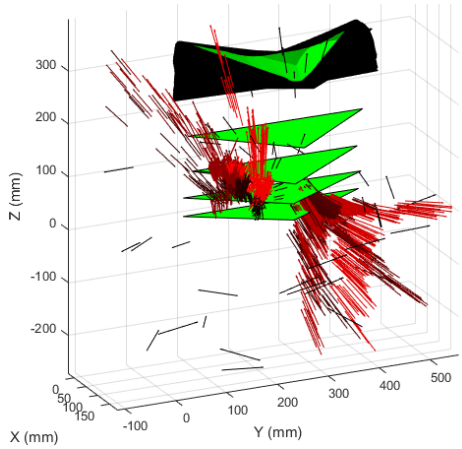
As previously stated, the screw analysis was used to determine the piece-wise motion profile of the torso. To achieve this, screws for each set of markers were determined with respect to the preceding set (ie. L4 with respect to S1, L1 with respect to L4, etc.). As seen in the raw results for the motion of the L4 set in Figure 3.10a, the complete set of screws varied in position and direction, and contained outliers. It can also be seen that multiple specific clusters occurred. If the subject's scoliotic curve is concave left in the lower region of the spine, the lower region of the spine is already bending to the left so the patient performing lateral bending to the left requires the upper region of the spine to perform the motion. Similarly, performing lateral bending to the right requires the lower spine to perform the motion. This likely caused the resulting clusters.

Data-based density spatial clustering of applications with noise (DBSCAN) was

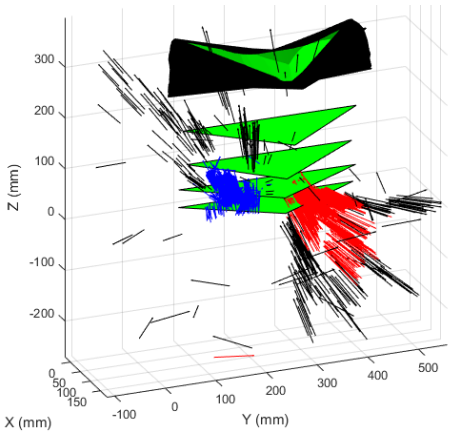
used to isolate these clusters and eliminate outliers. DBSCAN requires:

- data set of m objects by n variables.
- minimum number of objects considered as a cluster.
- neighboring radius.

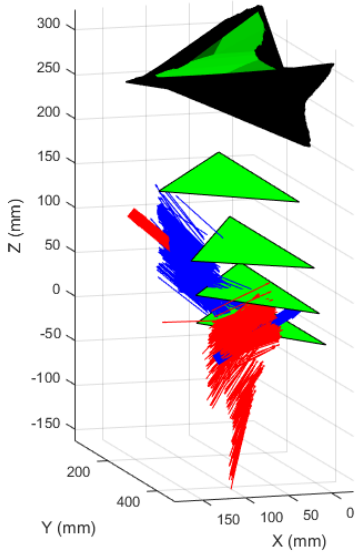
The objects were the time steps of data, and the variables were the x, y, and z coordinates of the points locating the screw combined with the x-, y-, and z-directions of screw directions. The minimum number of objects was between 25 and 100, and the neighboring radius was between 5 and 10, both dependent on the data set. Figure 3.10b shows the results of DBSCAN for this set of data, where the blue and red screws are two clusters and the blacks screws are outliers. An average screw was calculated for each cluster. Both $\vec{q}$ and $\vec{\omega}$ were averaged for each cluster (Figure 3.10c). The averages are plotted independently in Figure 3.10d.



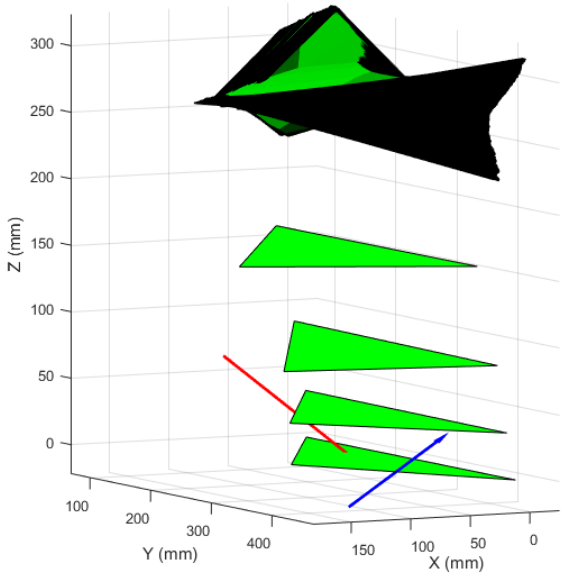
(a) Step 1 - with all screws



(b) Step 2 - clustered using DBSCAN method



(c) Step 3 - clustered screws plotted with average screw for each cluster



(d) Step 4 - only averages from each cluster

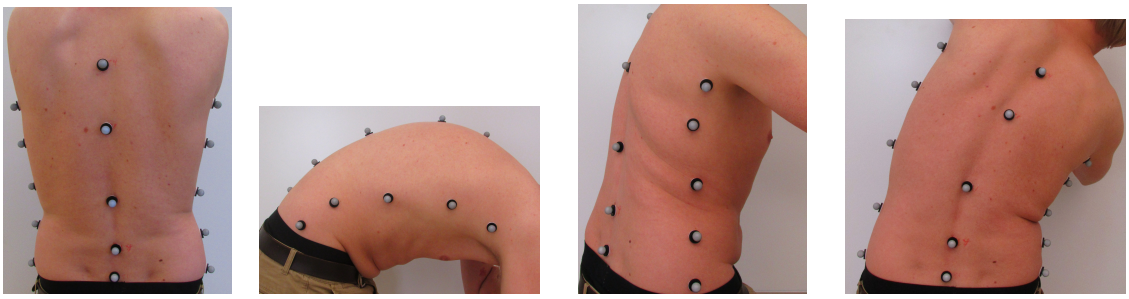
Figure 3.10: Depiction of motion analysis for the example of lateral bending motion. Screws represent the motion of the L4 set with respect to the S1 set.

3.2.3 Motion Capture Results

As previous stated, marker sets were placed at the S1, L4, L1, T8, and T4 vertebrae. To place the markers, the L4 vertebra was identified as the vertebra in line with the

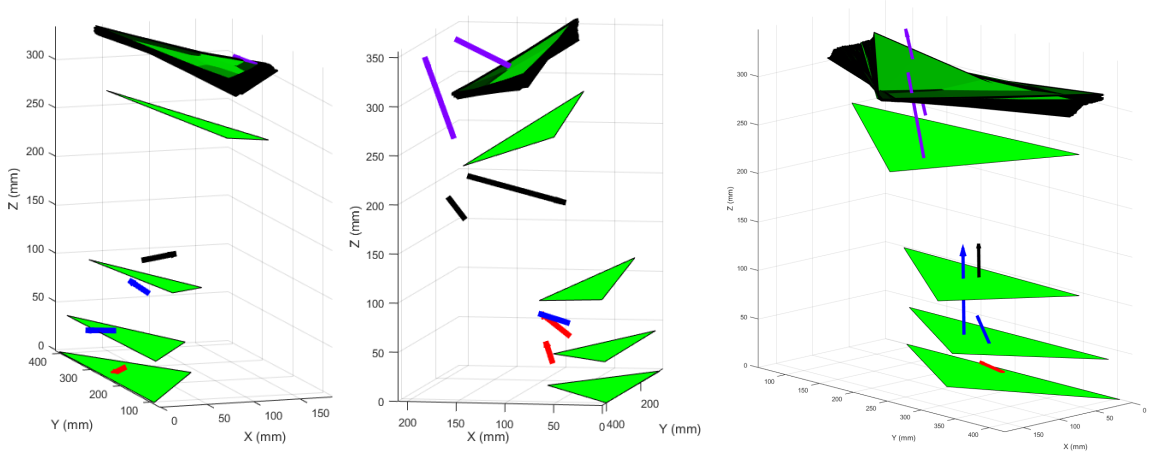
iliac crest [36]. Palpation was used to locate the L4 vertebra, and then to count up the spine to find the other vertebrae. The markers were fixed to the torso using double-side adhesive foam dots. Motion capture was taken while the subject was instructed to perform flexion, lateral bending, and twist three times for each motion. The position coordinates of the markers were exported from Nexus, the motion capture program, into CSV files. These files were input into MATLAB, where the previous analysis was performed.

Figure 3.11 displays the three motions being performed, and the resulting screws for each motion can be seen in Figures 3.12 and 3.13. One major assumption was made during this analysis: each primary motion of the torso can be defined by a single average screw. This assumption is not true, but is necessary to be able to complete design. The screws describing the motion drifted throughout subject's motion. In design, the methodology sought to align the mechanism's rotation axis with a specific requirement axis that described the motion. This average screw simplification was made so that mechanism design could be performed in this way.



(a) **(b)** **(c)** **(d)**

Figure 3.11: Subject performing the three primary motions. (a) Stationary (b) Flexion (c) Twist (d) Lateral Bending



(a) Sagittal Bending **(b)** Lateral Bending **(c)** Twist

Figure 3.12: All average screws of clusters for 3 motions, where the red arrow represents the L4-S1 set, blue the L1-L4, black the T8-L1, and purple the T4-T8.

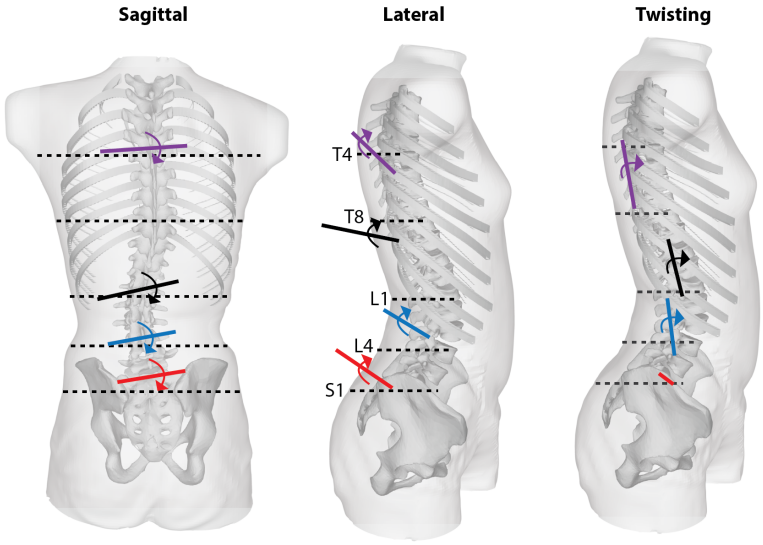
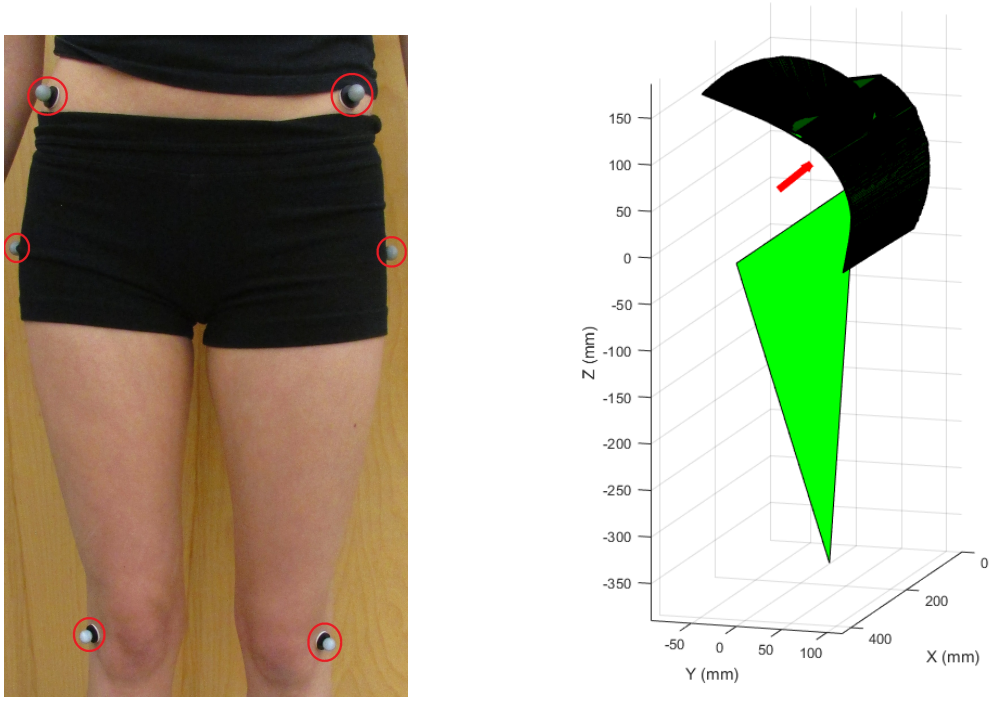


Figure 3.13: Kinematic description of the torso's motion using screws.

It was noticed during testing that a large amount of motion in flexion is performed at the hip. Braces generally do cover this area, so a motion capture analysis was performed to characterize the hip bending motion. Markers, seen in Figure 3.14a were placed on the left and right trochanter, left and right knees, left and right of the

pelvis, and on the S1 vertebrae. The analyzed motion can be seen in Figure 3.14b, where the trochanter markers and one knee marker produced the stationary set, and the hip and S1 markers made up the moving set.



(a) **(b)**

Figure 3.14: (a) Marker placement (circled in red) for hip motion capture. S1 marker is not pictured. (b) Resulting screw of motion

In addition to these general images, quantitative results were also determined. These quantitative results described the outcomes stated at the beginning of this section: the full range of motion of the patient, locations and directions of screw axes, and the piecewise characterization. The piecewise characterization was first explored. In Table 3.2, the motion angle is the angle of rotation at the subject's maximum deflection. For the percent of maximum motion angle, each motion angle was divided by the maximum motion angle for that specific motion. This provides insight into where the motions occur. For each motion, the following were found:

- *Flexion*: Most spinal bending occurs within S1 to T8. Looking at Figure 3.13, it is seen that the screw for the L1-L4 set is near the L3 vertibra. The large magnitude of the hip bending is notable because it is substantially greater than the other components that make up flexion. This motion occurs below the S1 vertibra but above the trochanter.
- *Lateral Bending*: Most bending occurs within L4 to T8 and the primary screw axis is near the T9 vertibra.
- *Twist*: Most twist occurs in the L4 to T4 and the screws line up with the spine.

In general, the subject primarily used their lumbar region for sagittal bending, the thoracic spine for twist, and a combination of lumbar and thoracic for lateral bending. These results of the motion analysis provide one of main contributions of this thesis.

Table 3.2: Magnitude of motion as characterized by the screw analysis.

Motion	**Spinal Section**	**Motion Angle (deg)**	**% of Max Motion Angle**
Flexion	Hip Bending	76.94	
	S1-L4	16.65	58%
	L4-L1	28.78	100%
	L1-T8	19.32	67%
	T8-T4	5.70	20%
Twist	S1-L4	5.17	18%
	L4-L1	16.59	57%
	L1-T8	29.33	100%
	T8-T4	11.69	40%
Lateral Bending	S1-L4	3.93	16%
	L4-L1	19.83	82%
	L1-T8	24.25	100%
	T8-T4	4.86	20%

Because the case study provided a very small area where mechanisms could be placed, it was decided that the brace would focus on two motions, flexion and lateral

bending. This decision was motivated by the hypothesis that it would be easiest to design for these two motions and the fact that, in general, twist requires the least motion for each ADL. The three overall outcomes of the kinematic study, which define the requirements for those two motions, are presented in Table 3.3. The rotation requirements are the percentage of the subject's maximum motion, as stated in Section 3.2.1. The general and detail requirements will be used in the topology and dimensional design, respectively.

Table 3.3: List of requirements derived from motion capture.

	General Requirements	Detail Requirements
Flexion	• Rotation axis lateral to torso • Rotation axis between pelvis and trochanter	• $\vec{\omega} = [.99, -0.04, 0.08]$ • $\vec{q} = [0, 44, 100]$ • rotation $\theta = 30.8\,\text{deg}$
Lateral Bending	• Rotation axis back to front of torso • Rotation axis near T9	• $\vec{\omega} = [0, 0.97, 0.25]$ • $\vec{q} = [-.01, 0, 349]$ • rotation $\theta = 7.28\,\text{deg}$

3.2.4 Possible Sources of Error

There were three main sources of error in this kinematic analysis: marker placement, the rigid body assumption, and variability of motion. Placing the markers required feeling the spine through the skin and counting the vertebrae. In general, the L4 vertibra aligns with the top of the pelvis. This was used as a baseline, but placing the markers on the spine can be difficult, even for the trained professional [37]. Missing a vertebra or miscounting was entirely possible.

Error can also stem from the subject's skin moving relative to the vertebrae and the rest of the bony structure when performing a motion. This inherently proved that the assumed set of markers is not a rigid body, but the rigid body assumption

was required for the usage of screw theory, so the raw motion data was modified. Neglecting the relative motion caused error but was necessary to perform the analysis.

Lastly, a person's motion path can vary every time that they perform a motion because the spine can bend in many different configurations. The spine is made of 17 vertebrae in the lumbar and thoracic region, all of which can move relative to each other. The motion study completed for this design however produced very specific rotation axes and suggested specific regions where motions occurs. It was considered in design that the brace could still allow the patient to perform a motion even if the brace did not exactly match the specified requirement. The patient can modify how they perform a motion to comply with the brace. That being said, it was the goal of the compliant brace to closely follow the natural motion of the torso.

3.2.5 Conclusion

Patient-specific kinematic requirements were developed to use in the design of the brace. This is a new type of study that could be used for similar applications, as was done to characterize the motion of hip bending. The regions of the spine that are predominantly responsible for of each primary motion were noted such that the information can be used to locate mechanisms. The maximum of each primary motion that the subject could complete was studied. In combination with data from the literature, the amount of each primary rotation that brace will be required to complete was determined. Together, these results will make up the kinematic requirements for the brace.

3.3 Force Requirements

3.3.1 Correction Paradigm

Introducing compliant mechanisms to the scoliosis bracing problem brings about a complex question of whether to design the brace with high stiffness (displacement control) or low stiffness (force controlled). This continuum is defined by

$$\vec{u} = C\vec{f} \tag{3.35}$$

where u is the displacement, C is the compliance matrix, and f the force. The brace must be designed in a hyper-corrected state such that the mechanisms are loaded and force is applied when it is fit on the torso. The degree of hyper-correction of the brace will define where on the continuum the final design will be found. A high stiffness causes the force of the brace to be sensitive to change in displacement, meaning that the force will decrease greatly if the patient moves their body away from the force. A low stiffness causes the displacement to be sensitive to change in force. A relatively constant force can be applied using low stiffness because the force will change a minimal amount with change in displacement.

This brace will seek to achieve a high stiffness, displacement control. The motivation for this paradigm is:

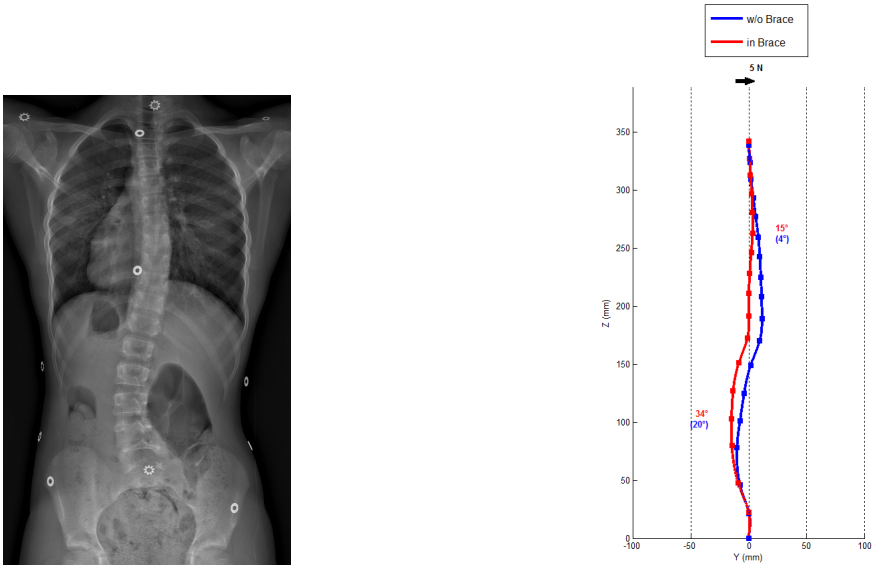
- The most successful braces currently on the market, the Boston and Cheneau braces, are rigid plastic that use the displacement correction paradigm. A well-proven paradigm will be a good place to start for designing a new type of brace.
- The off-patient, zero-force equilibrium of the brace will require less hyper-

correction than the force-controlled brace, because a higher stiffness allows the mechanism to reach higher forces with less displacement. This should provide for easier and more plausible mechanism design.

3.3.2 Integrating Forces

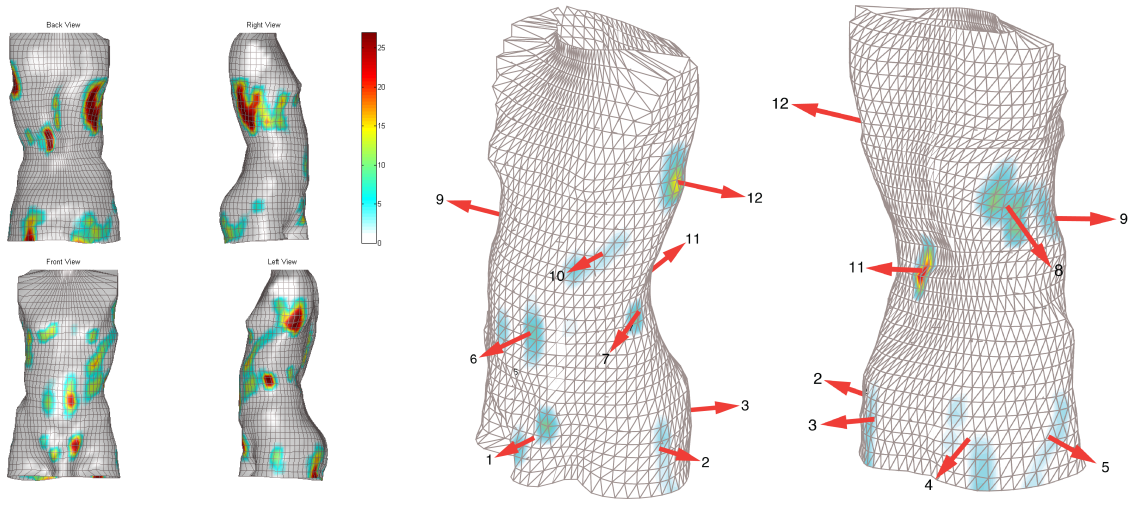
As stated in the Literature Review, most braces focus on displacement control for correction. Therefore, the literature regarding the biomechanics of scoliosis bracing contains minimal information pertaining to the forces needed for correction. More information was gained from collaboration with Laboratoire d'innovations CAO en génie orthopédique at the École Polytechnique de Montréal. Our collaborators provided a mapping of pressures caused by a brace on a patient. The patient's X-ray (Figure 3.15) shows an S-curve with initial Cobb angles of 34° and 15°. The patient was similar in size to the subject used in the kinematic analysis, as well as having similar scoliotic curves.

From the mapping (Figure 3.16a), forces were integrated over the area. This mapping was of the pressure applied to the torso with the spine in the corrected state. The surface composed of triangular planes. For each triangle, the pressure was multiplied by the area of the triangle and assumed to act in the direction normal to the triangle at the triangle's centroid. Twelve areas of pressures were identified and the total force was summed for each patch, resulting in the twelve forces seen in Figure 3.16b. The magnitudes of the forces are listed in Table 3.4. An assumption that these pressures can be replaced with point loads was used for design. The locations, magnitudes, and directions of these forces are taken as the requirements that make up the corrective force system.



(a) **(b)**

Figure 3.15: (a) X-ray showing the curve of the patient used to derive force requirements. (b) Alignment of vertebrae with and without the brace.



(a) **(b)**

Figure 3.16: (a) Pressure map of brace forces from collaborators (b) derived force vectors.

Forces 7, 8, 9, and 12 are the primary correction forces as they occur in the lateral direction and oppose each other. The magnitudes of these forces are larger than the 20-40N found in literature, but as was stated, the literature was not reliable. Our collaborators mentioned that these pressures, and therefore forces, vary with the

tension in the brace straps. Although the magnitudes are different, the number and location of forces, providing a 4-point correctional system with forces at or below the apexes of the curve, is consistent with literature.

Table 3.4: List of forces results from integration of pressures.

		Force Components (N)		
Force #	**Force Magnitude (N)**	x (front-back)	y (lateral)	z (vertical)
1	31	31	1.2	1.8
2	23	-4.9	-22	-2.5
3	31	24	-18	-0.1
4	32	30	10	0.4
5	36	23	27	-1.6
6	44	-44	-1.2	0.0
7	40	-36	-18	-1.7
8	76	49	55	16
9	38	-8.1	37	-2.3
10	22	-21	-4.8	-1.1
11	39	38	0.0	-9.4
12	50	-0.3	-50	-4.2

3.3.3 Possible Sources of Error

There is possibility for error in the magnitude, direction, and location of forces. The direction is likely most affected because it relies on the scanned surface of the torso, which may not accurately represent the contact between the brace and body. A better option may be to look at the pressured mapped to the brace's surface, as this is a smooth, continuous surface. The scanned surface may also not represent an accurate surface area, in which case the magnitudes of force would be incorrect.

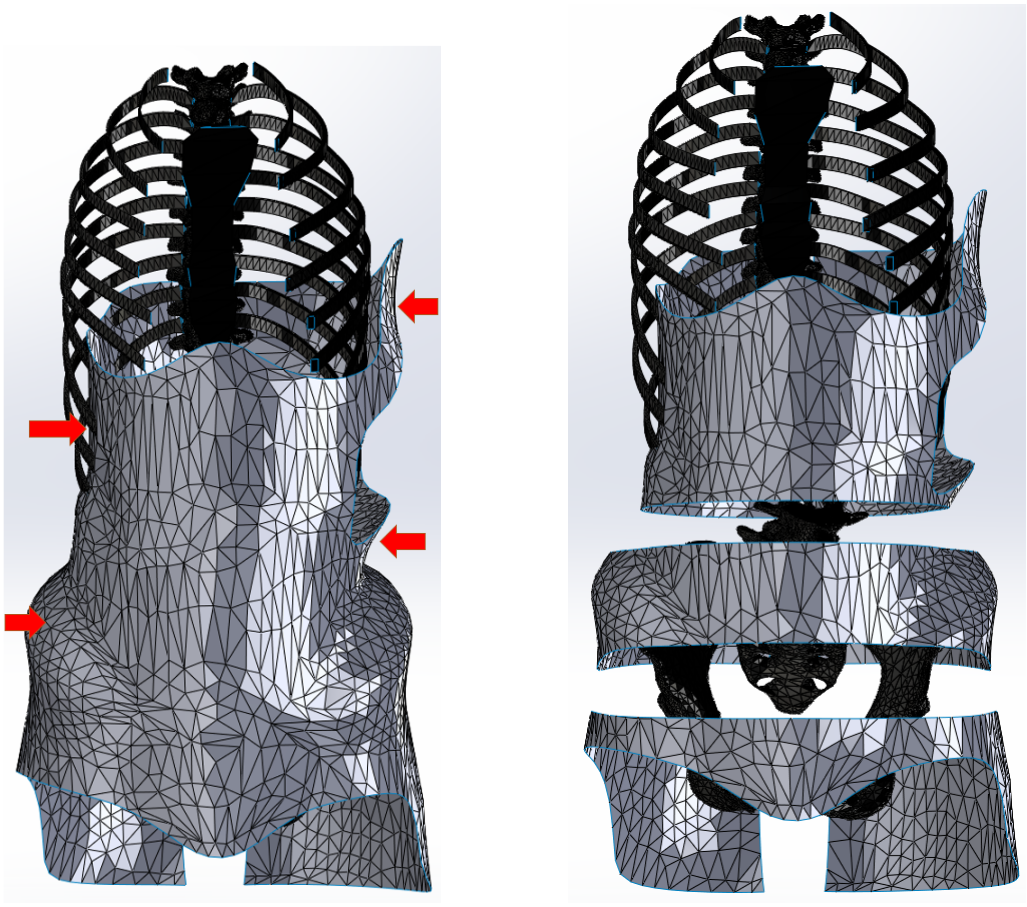
Even with this possible error, these forces were used for the design of the flexible scoliosis brace because they present the most complete understanding of the correctional force system. In the future, an analysis of pressure maps from many patients

could draw a correlation to the general amount of force needed for correction.

3.3.4 Designing Brace Pieces

As stated in Chapter 2.3, the topology design first entailed defining rigid brace pieces through which corrective loads were imparted. Subsequently, compliant flexure mechanisms were selected to match the motion and force requirements. Material from a successful rigid brace was removed to produce the three pieces that were to be connected by compliant mechanisms. Once the brace pieces were known, the requirement compliance ellipsoids were derived with respect to the motion and possible mechanism location. The requirement ellipsoids were compared to the building block library, and appropriate mechanisms were chosen.

The goal of the flexible brace is to replicate the corrective forces from the original brace because it is known that the original brace provided a successful treatment. Therefore, the brace needs to be able to apply the primary correction forces, noted in red in Figure 3.17a, in the same locations. To ensure that the forces would be transmitted to the same locations as the original, segments of the brace where larger pressures were originally applied were maintained while other segments were removed to allow for motion. Mechanisms were placed where segments of the brace were removed. The brace was split into the three pieces seen in Figure 3.17b from the original design in Figure 3.17a. Three of the primary forces are applied by the upper brace piece, and the other primary force is applied by the middle brace piece. The lower piece accounts for reaction forces.



(a) **(b)**

Figure 3.17: (a) Original brace design and (b) modified brace pieces

Now that it is known which brace pieces need to apply which forces, the forces in Table 3.4 can be grouped and summed. Table 3.5 states which forces were summed for each brace piece and the resulting force sums. Furthermore, the resulting sums of moments, caused by the forces on each brace piece, were reported.

Table 3.5: Summed forces and resulting moments from brace pieces.

		Summed Forces (N)			**Resulting Moment (N-mm)**		
Piece	**Forces**	x	y	z	x	y	z
Low	1, 2, 3, 4, 5	42	-2.2	-1.9	0.3	-1.2	0.6
Middle	6, 7, 11	-41	-19	-11	-0.2	0.3	0.2
Upper	8, 9, 10, 12	19	37	8.6	-0.9	0.3	0.4

For dimensional design of the mechanisms, the interaction forces between brace pieces are needed. It should be noted that the forces did not sum to zero. This error occured because the force integration method is a estimate with estimated directions. To overcome this, it was assumed that the forces on the low piece result from the forces on the middle and upper pieces, and therefore could be calculated as a resultant. A static sum of forces was performed on the brace pieces with that assumption. The forces that the middle piece needed to apply to the upper piece, F_{um} was the same as what the brace needed to impart on the body. The force that the low piece needed to apply to the middle piece, F_{ml}, was the force that middle brace imparts on the body subtracted from F_{um}. This was also applied to moments, where the moments in Table 3.5 were calculated as the sum of the cross product of each force and its distance vector to a common point. In summary,

$$F_{um} = F_{upper} \tag{3.36}$$

$$F_{ml} = F_{um} - F_{middle} \tag{3.37}$$

$$M_{um} = M_{upper} \tag{3.38}$$

$$M_{ml} = M_{um} - M_{middle} \tag{3.39}$$

where the F_{upper}, F_{middle}, M_{uppper}, and M_{middle} are drawn from Table 3.5. The results of this calculation, which are the required mechanisms forces, are noted in Table 3.6.

Table 3.6: Required mechanism forces.

	Resultant Force (N)			**Resultant Moment (N-mm)**		
Forces	x	y	z	x	y	z
Upper-Middle	19	37	8.6	-0.9	0.3	0.4
Middle-Lower	-22	18	-2.5	-1.2	0.7	0.6

3.4 Conclusion

Both the motion and force requirements, determined for the specific case study used to design the compliant brace, are reiterated in Tables 3.7 and 3.8, respectively. The motion requirements were identified through a motion study which provided both general requirements for topology design and specific requirements for dimensional design. This characterization of the spine provided a large contribution from this thesis because no previous studied have characterized the motion in a spatial manner. The force requirements were determined by integrating a pressure map of the pressure applied to the torso by the spine. Although the literature provided suggestion for the location of the forces, it did not provide information on the necessary magnitudes for correction. Therefore, this method used to determine force magnitudes also proves to be a contribution of this thesis. With these requirements defined, the design of the brace can now begin.

Table 3.7: Kinematics Requirements

	General Requirements	Detail Requirements
Flexion	• Rotation axis lateral to torso • Rotation axis between pelvis and trochanter	• $\vec{\omega} = [.99, -0.04, 0.08]$ • $\vec{q} = [0, 44, 100]$ • rotation $\theta = 30.8\,\text{deg}$
Lateral Bending	• Rotation axis back to front of torso • Rotation axis near T9	• $\vec{\omega} = [0, 0.97, 0.25]$ • $\vec{q} = [-.01, 0, 349]$ • rotation $\theta = 7.28\,\text{deg}$

Table 3.8: Force Requirements

	Resultant Force (N)			**Resultant Moment (N-mm)**		
Forces	x	y	z	x	y	z
Upper-Middle	19	37	8.6	-0.9	0.3	0.4
Middle-Lower	-22	18	-2.5	-1.2	0.7	0.6

Chapter 4

Compliant Mechanism Design

The design of compliant mechanisms requires motion and force requirements, which were found in the previous chapter. The general methodology used to design the brace and define compliant mechanisms is depicted in Figure 4.1. The design of the brace was performed in two parts: topology design and dimensional design. The The first step of the topology design was to separate a successful rigid brace into segments to be connected by compliant mechanisms. Once these brace pieces were designed, mechanisms were chosen that allowed the desired motion. Few methods exist that could have been used to choose the ideal mechanisms in this spatial design problem, so a 2D method, the Building Block Approach, was expanded for 3D, spatial usage. In this expansion, a library of building block (compliant joints) each with a visual characterization was created. From this library, compliant joints were chosen to perform a specific motion with respect to the requirements of the brace. This concluded the topology design portion of the process.

The dimensional design was comprised of two steps: aligning the mechanism's

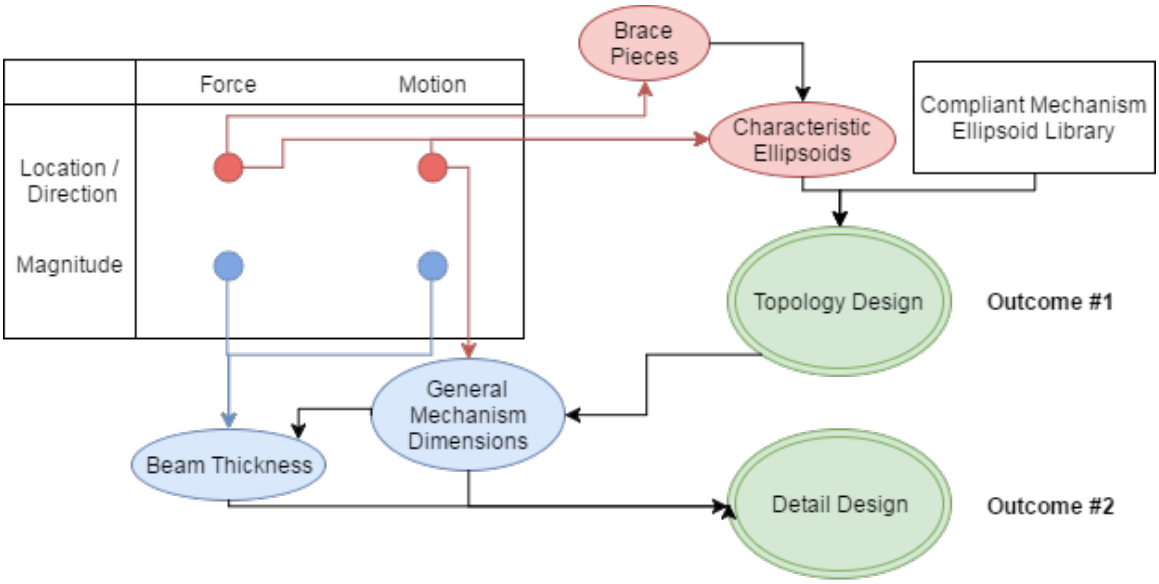


Figure 4.1: Flow chart of process used to design the scoliosis brace with compliant mechanisms.

rotation axis to the requirement and determining the ideal beam thickness to allow the most motion while applying the necessary correctional force system. The first step utilized a tool called the eigentwist to define the primary axis of rotation of the mechanisms. All dimensions of the mechanisms but the beam thicknesses affected the location of the eigentwist, so these dimension were used to align the eigentwist with the requirement screw. The last step was to define the beam thickness. The system was modeled in a finite element software, and it was found that the mechanisms were capable of achieving the greatest range of motion when the beams were the thickest without reaching their fatigue strength when 50 Newtons of load was applied. This criterion was used to determine the ideal beam thickness for each mechanism. The brace was fully designed when these steps were completed.

The contributions from this chapter are the expansion of the Building Block Approach from a 2D to 3D and the use of the eigentwist to align mechanism with their requirement. The steps of this process can be seen in Figure 4.1 with the requirements

that were used for each step.

4.1 Building Block Approach Expansion

The key contribution of this thesis to the field of compliant mechanisms is the expansion of the Building Block Approach into a 3D synthesis method. Other synthesis methods, as previously presented, are not ideal for this design problem. The furthering of this method will allow for the designer to easily compare building blocks to a design problem characterization in order to find possible solutions. This section will review the derivation of the compliance ellipsoid originally developed for compliant mechanisms by Kim [16] and produce a library of compliant mechanisms characterized with compliance ellipsoids. Then, this expanded method is utilized in the design of the scoliosis brace. The major contribution of this section is the expansion of the previously developed Builidng Block Approach for use in spatial problems.

4.1.1 Ellipse to Ellipsoid

Previously used in robotic applications, Kim introduced the compliance ellipse for usage in compliant mechanism design through the development of the Building Block Approach [16]. The compliance ellipse is a planar visual representation of the compliance of a mechanism derived from the compliance matrix, C, in

$$\vec{u} = \mathbf{C}\vec{f} \tag{4.1}$$

where

$$\mathbf{C} = \mathbf{K}^{-1} \tag{4.2}$$

such that

$$\vec{f} = \mathbf{K}\vec{u}. \tag{4.3}$$

For a general mechanism considered in a plane (Figure 4.2), $\vec{f}$ represents the force applied at some point, P, on the mechanism and $\vec{u}$ represents the displacement of P. The figure also notes that the mechanism is constrained in all direction at Ω.

These constraints result in the problem being represented by a 3x3 compliance matrix where Eq. 4.1 can be written as

$$\begin{bmatrix} u_x \\ u_y \\ \theta \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix}. \tag{4.4}$$

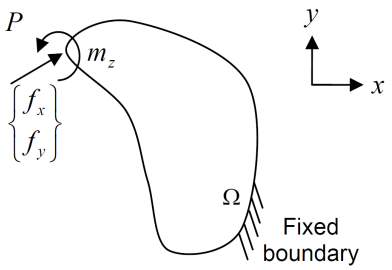


Figure 4.2: General representation of mechanism with applied load [16].

The top, left 2x2 of the **C** represents the translational motion of P. The eigenvalue problem,

$$\lambda\vec{\xi} = C_{2x2}\vec{\xi} = \vec{u}, \tag{4.5}$$

can be decomposed to define the ellipsoid. λ is the eigenvalue and $\vec{\xi}$ is the eigenvector. The eigenvectors represent the semi-axes of the ellipsoid. The eigenvector with the larger corresponding eigenvalue represents the primary compliance vector for translation (PCV_t), while the eigenvector with the smaller corresponding eigenvalue represents the primary stiffness vector for translation (PSV_t). These vectors constitute the semi-major and semi-minor axes of the compliance ellipse as would result if a unit circle of force was applied to the mechanism. This is seen in Figure 4.3.

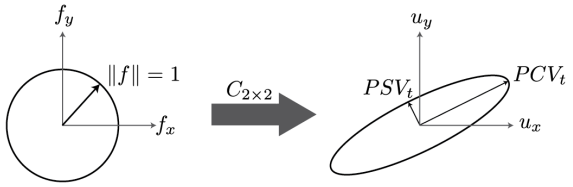


Figure 4.3: Mapping of a unit circle of force to a mechanism [17].

Analogously, a unit sphere of force can be applied to result in a compliance ellipsoid, as seen in Figure 4.4 [17]. In the spatial case, C is the 6x6 matrix,

$$\mathbf{C} = \begin{bmatrix} C_A & C_B \\ C_B & C_D \end{bmatrix}, \tag{4.6}$$

where C_A, C_B, and C_D are all 3x3 matrices. In addition the force and displacement vectors are represented by

$$\vec{f} = \begin{bmatrix} f_x & f_y & f_z & m_x & m_y & m_z \end{bmatrix}^T \tag{4.7}$$

and

$$\vec{u} = \begin{bmatrix} u_x & u_y & u_z & \theta_x & \theta_y & \theta_z \end{bmatrix}^T, \tag{4.8}$$

respectively. Both translation and rotational ellipsoids can be created for the spatial case. The semi-axes of the translation compliance ellipsoid are derived from the eigenvalue decomposition of C_A for the problem,

$$\lambda\vec{\xi} = C_A\vec{\xi} = \vec{u}. \tag{4.9}$$

The semi-axis eigenvectors are noted as primary, secondary, and tertiary compliance vectors for translation (PCV_t, SCV_t, and TCV_t). PCV_t is the primary displacement vector and TCV_t is the primary constraint vector. The rotational compliance ellipsoid is derived from the eigenvalue decomposition of C_D. Similarly, the eigenvectors are noted as PCV_r, SCV_r, and TCV_r. The combination of the translation and rotational compliance ellipsoids produce a general representation of the force-deflection characteristics of a compliant joint.

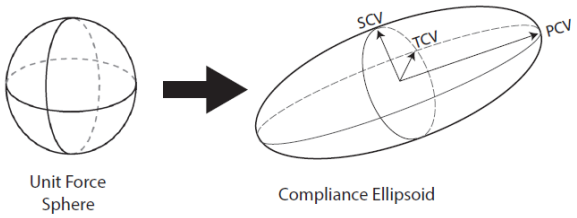


Figure 4.4: Mapping of a unit sphere of force to a mechanism [17].

Limits of Ellipsoid Model

The ellipsoid representation does not fully represent the motion of a compliant joint for two reasons. First, the representation does not account for the coupling that occurs between rotational and translational displacement. The coupling is accounted for by C_B in the compliance matrix, Eq. 4.6. This coupling was not necessary for a general understand of compliant joints, but was accounted for in the dimensional

design of the mechanisms. The second limitation is that this is a linear representation so the ellipsoids are only guaranteed to represent the motion at the rest configuration of the joint. Once the joint is displaced, the characteristics of the joint can change. Even with these two limitations, the Spatial Building Block Approach provides a useful method for choosing mechanisms for a specific design problem.

4.1.2 Library of Building Blocks

With this knowledge, the characterizations of compliant joints were generated and implemented in a library of joints. To characterize a joint, the only requirement is finding the compliance matrix. The method used to determine the compliance matrix for a joint can be seen in Appendix A.

Examples of building blocks and their characteristic ellipsoids are seen in Table 4.1. For each of the beam-based mechanisms, the planar thickness and beam thickness are 3 mm and 2.33 mm, respectively. Consider the cartwheel hinge and the parallel beam joints. Their linear translation ellipsoids are similar in that they both allow out-of-plane motion and motion in the their primary actuation directions (x-direction). Both also allow little compression in the y directions. The primary difference is in the rotational ellipsoid where the semi-axis in the z-direction is very small (restricts rotation) for the parallel beam joint, but larger for the cartwheel hinge. The cross-pivot hinge has even more relative flexibility in the z-rotational direction, as can be seen by its rotational ellipsoid.

Table 4.1: Library of flexures characterized by compliance ellipsoids.

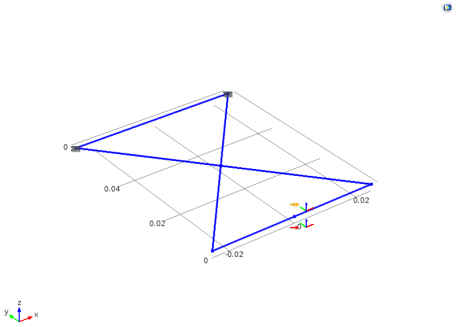
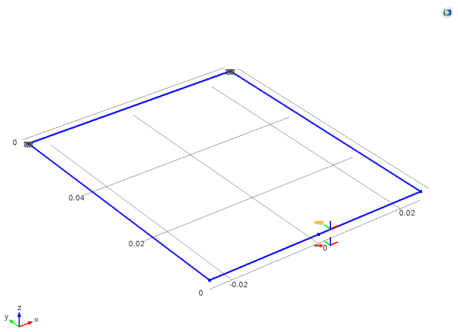
Mechanism	Ellipsoids
Cartwheel Hinge	
Parallel Beam	

Table 4.1: Library of flexures characterized by compliance ellipsoids.

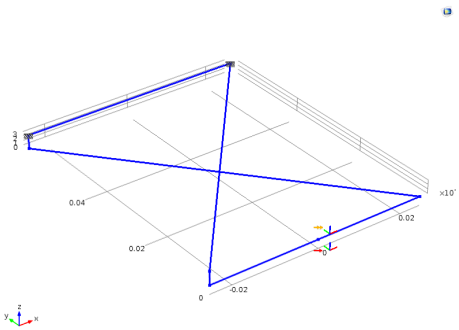
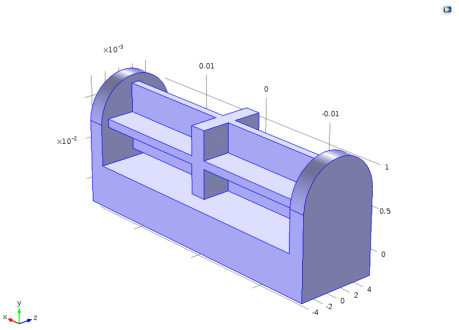
Mechanism	Ellipsoids
Cross-Pivot	
Cross Beam	

Table 4.1: Library of flexures characterized by compliance ellipsoids.

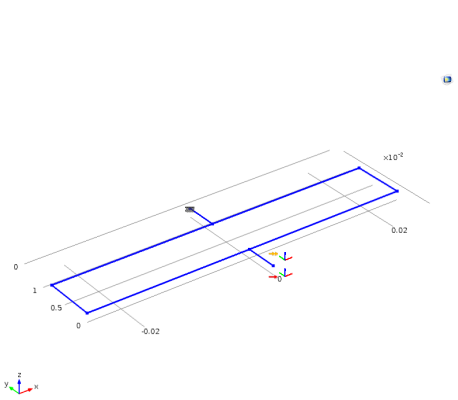
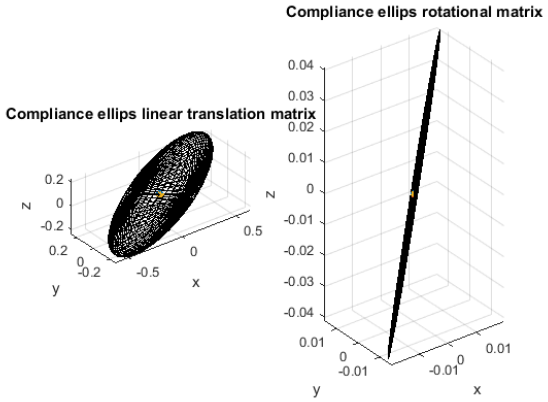
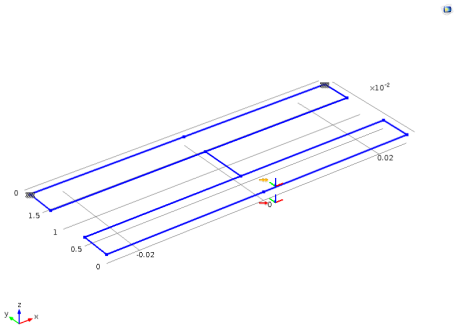
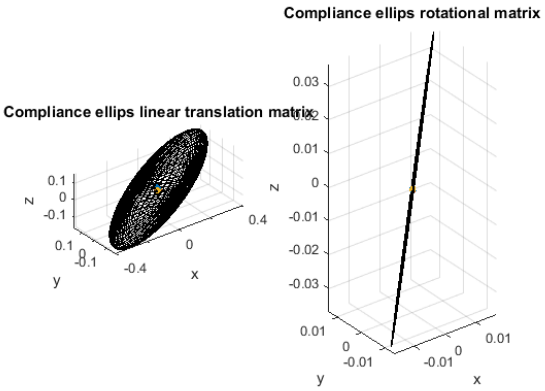
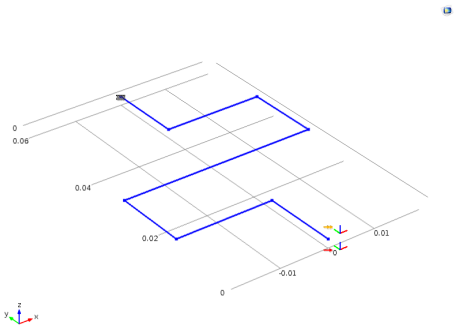
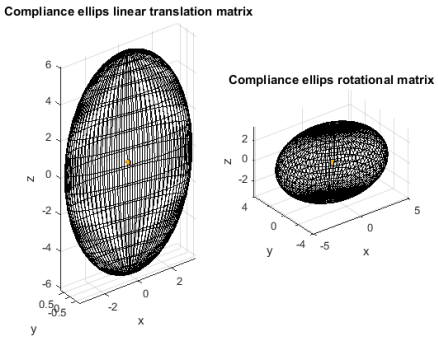
Mechanism	Ellipsoids
LET Outside	 
LET Inside	 

Table 4.1: Library of flexures characterized by compliance ellipsoids.

Mechanism	Ellipsoids
S-beam 	

4.1.3 Conclusion

The methodology for characterizing the translation and rotational freedom of building blocks using compliance ellipsoids such that it can be applied to new mechanisms was presented. Furthermore, the methodology was used to build a library of flexure type building blocks. If a problem is then characterized in the same way, with compliance ellipsoids, the characterization can be visually compared to the building blocks to find all possible solutions.

4.2 Topology Design using Building Blocks

4.2.1 Characteristic Requirement Ellipsoid Generation

In order to choose mechanisms by direct comparison of ellipsoids, the problem must be characterized using ellipsoids. Depending on where the mechanism is located relative to the torso, the required characteristic ellipsoids vary. If the plane of a mechanism is parallel to the axis, the distance away from the axis will determine whether rotation or extension is needed. Four locations (top of Figure 4.5) of mechanisms for flexion and lateral bending were considered: sides of the torso, back and front, two separate mechanisms on the back, and a mechanism that encompasses the torso. The first two options were the extremes: mechanism parallel to axis and mechanism orthogonal to axis. The third option provides a solution between each of these extremes. The torso encompassing mechanism was not considered for flexures, but is necessary to provide a full description of possibilities for mechanisms such as shells.

For twist, the configurations include: one mechanisms on each side of the torso, one mechanism on the back, one mechanism on the front, and a mechanism that encompasses the torso. The first three options represent the extremes for this motion: mechanisms far from the axis and mechanisms near the axis. Figures 4.5 through 4.7 depict possible mechanism locations and their resulting characteristic ellipsoids. The required kinematic degrees of freedom are also depicted by blue arrows for each mechanism location. To create the requirement ellipsoids, the semi-major axes were put in direction of required compliance and the semi-minor axes in directions of required stiffness.

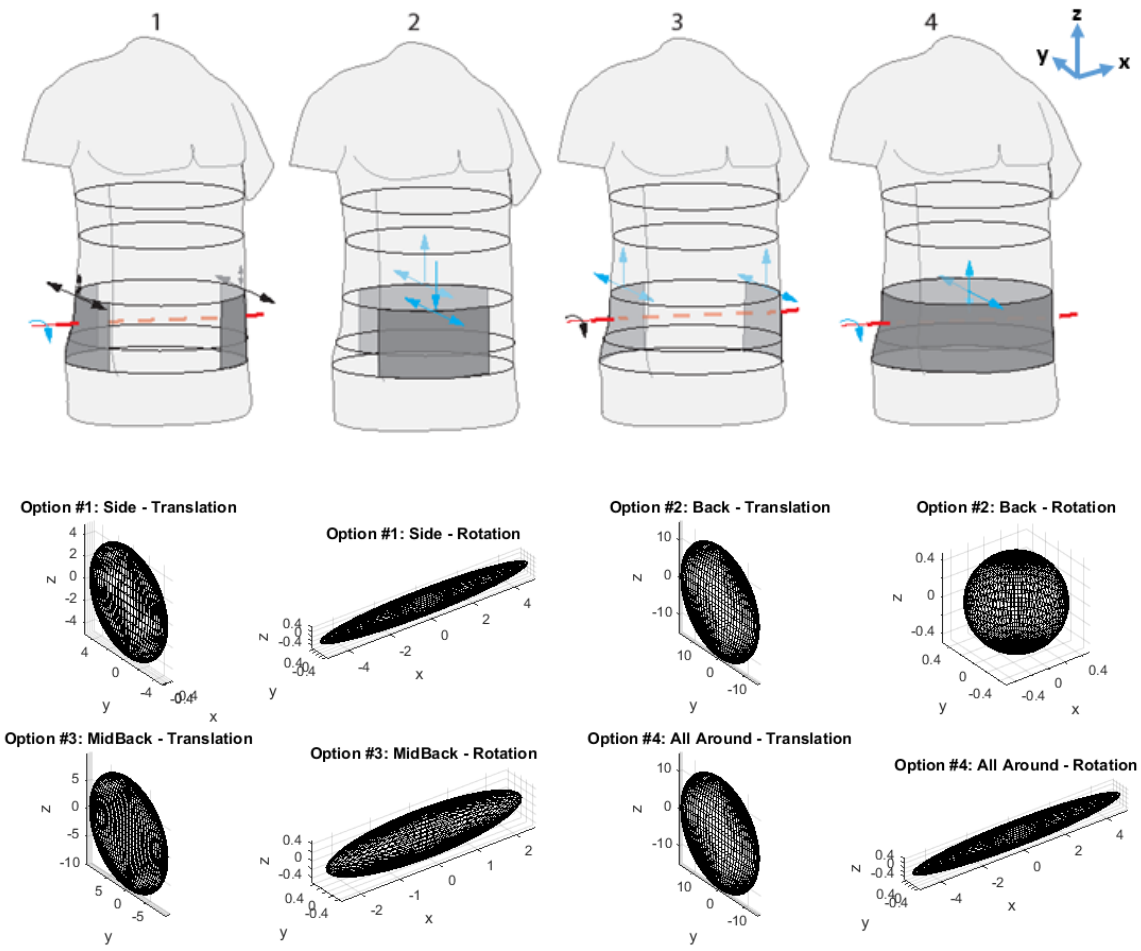


Figure 4.5: For *flexion*, (Top) possible mechanism locations and required DOF and (Bottom) associated characteristic ellipsoids.

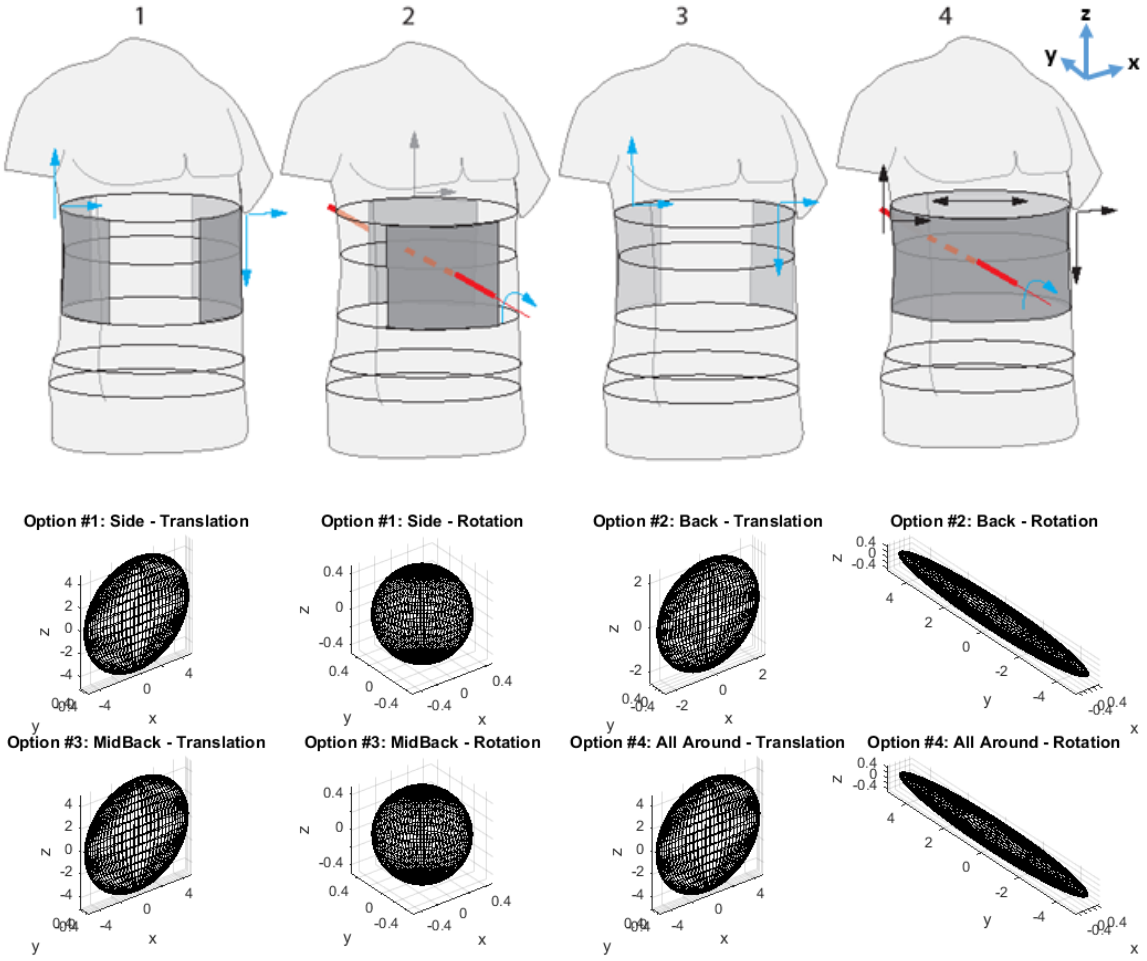


Figure 4.6: For *lateral bending*, (Top) possible mechanism locations and required DOF and (Bottom) associated characteristic ellipsoids.

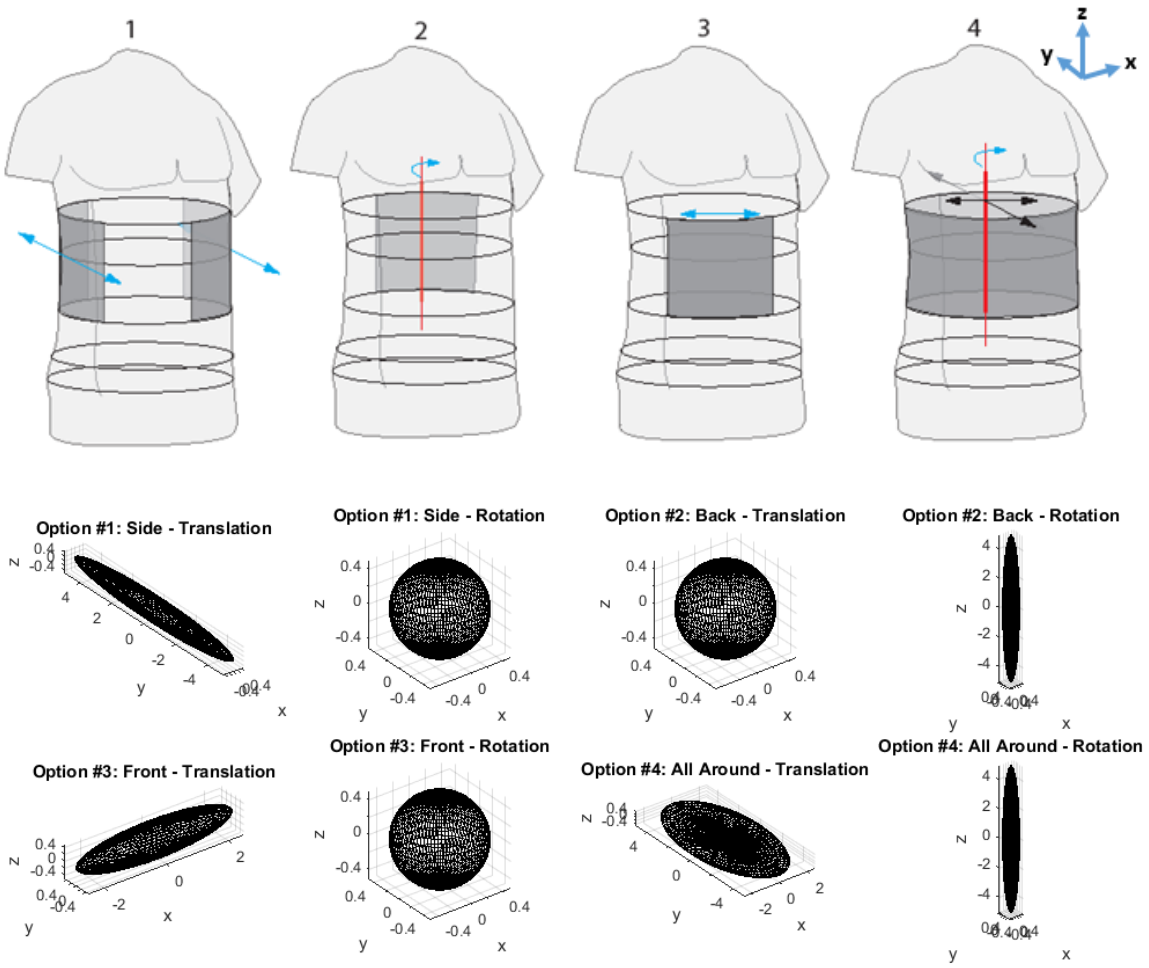


Figure 4.7: For *twist*, (Top) possible mechanism locations and required DOF and (Bottom) associated characteristic ellipsoids.

4.2.2 Building Block Selection

The kinematic requirement ellipsoids, combined with knowledge that the primary stiffness direction is lateral, were used to choose mechanisms from the building block library in Section 4.1.2. The compliance characteristics were the primary goal with the stiffness requirements further reducing the number of possible mechanisms.

Table 4.2 provided all options from a first review of the mechanism library. Flexion with mechanisms placed on the sides of the torso, Figure 4.5 Option #1, was considered. Note that for this configuration, the ellipsoids needed to be rotated 90° about the z-axis and then x-axis from the position that they are portrayed in the library to be parallel to the surface of the torso. The primary kinematic freedom requirement of flexion was rotation about a lateral axis. For this reason, mechanisms with PCV_r in that direction was ideal. Since the lateral stiffness was considered as the constraint direction, a TCV_t in the lateral direction was preferred. This included all but two mechanisms from the library. With further consideration of the ideal mechanism, the cartwheel hinge and cross-pivot were found to be preferred because they offered the least out-of-plane translational compliance. Of these two, the cross-pivot was the best option because the semi-axis in the out-of-plane direction was relatively smaller than that of the cartwheel hinge.

Table 4.2: Possible mechanism choices noted by red blocks. All 21 options (7 motions in 3 locations) are included for each motion.

	Flexion			Lateral Bending			Twist		
Mechanism Position	Sides	Back	Mid	Side	Back	Mid	Side	Back	Front
Cartwheel	■				■			■	
Parallel 2-Beam							■	■	■
Cross-Pivot	■	■							
Cross-Beam									
LET Outside	■	■	■	■				■	■
LET Inside	■	■	■	■				■	■
S-Beam	■	■	■						

For lateral bending, fewer building blocks were plausible because the required stiffness and compliance had a coupled relation. The mechanism needs to be relatively stiff in translation laterally, but compliant in rotation in that lateral direction, as

noted in Figure 4.8. Therefore, only three building blocks were ideal. Of the three, the cartwheel hinge was the best option because it prefers rotation to translation. The LET type joints, which would have to be placed on the sides of torso would have to compress but not succumb to out-of-plane motion, which they are not designed to do, as described by their compliance ellipsoids.

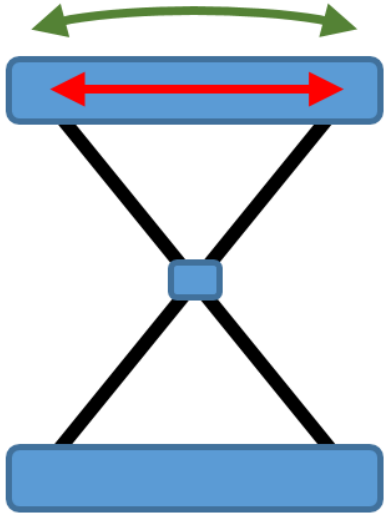


Figure 4.8: Cartwheel hinge with required in-plane translational stiffness (red) and in-plane rotational compliance (green).

4.2.3 Conclusion

The general design of the brace was determined in this section through application of the expanded building block approach. The cross-pivot and cartwheel hinge have been chosen for flexion and lateral bending, respectively. Both mechanisms were characterized by a primary axis of rotation in the out of plane direction. The cross-pivot was chosen to connect the lower and middle brace pieces because flexion predominantly occurred at the hips. A cross-pivot was placed on each side of the torso to create a rotation axis lateral to the torso. The cartwheel hinge connected the middle

and upper brace pieces because lateral bending occurred in the middle-upper region of the spine. These mechanisms were placed on the front and back of the torso to achieve rotation axis in the front to back direction

The application of the expanded building block approach was the primary contribution of this section. The building block approach was expanded from planar problems for use with spatial problems. The expansion developed ellipsoids that describe the general translational and rotational compliance characteristics of a joint. The ellipsoids were used to characterize a library of joints (building blocks). From this library, mechanisms were chosen for the scoliosis brace problem resulting in the general topology design of the brace.

4.3 Dimensional Design using the Eigentwist Decomposition

The purpose of dimensional design was (1) to define the dimensions of the mechanisms such that they mimic the natural motion of the torso as closely as possible and (2) to determine the optimal thickness of the beams that allows for the most motion while still applying the necessary correctional force system. A tool called the eigentwist was used to accomplish the first part of dimensional design. The eigentwist was first introduced by Lipkin and Patterson to characterize robot elasticity [38]. In this thesis, the eigentwist was used to identify the primary axis of rotation of a mechanism. The eigentwist, which can be calculated through a modified eigenvalue decomposition of the compliance matrix, is a description of instantaneous velocity which itself represents a compliant mechanism's preferred direction of spatial mo-

tion. Once determined, the eigentwist was decomposed into screw parameters. This eigentwist, represented by screw parameters, was compared to the requirement screw determined from the motion capture. Comparing these allows for the alignment of the mechanism's preferred axis of rotation with the kinematic requirement screw. The eigentwist has not previously been used for this purpose. The final step is to optimize the beam thickness to produce the designed force system. Material is one of main contributors to the performance of compliance mechanisms, so it is also considered in this section.

4.3.1 Screw Alignment Using The Eigentwist

The eigentwist was originally introduced by Lipkin and Patterson for the modeling of robot elasticity [38, 39, 40]. In robotics, the source of elasticity, which is unwanted, is compliance in the joints, but in the application of compliant mechanisms, the elasticity originates by design from flexible members. Lipkin and Patterson developed the representations from screw theory such that translational and rotational displacement can be represented by an eigentwist,

$$\vec{T} = \begin{bmatrix} u_1 & u_2 & u_3 & \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T = \begin{bmatrix} \vec{\delta} \\ \vec{\gamma} \end{bmatrix} \tag{4.10}$$

while linear forces and moment couples are represented by

$$\vec{\omega} = \begin{bmatrix} f_1 & f_2 & f_3 & \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T = \begin{bmatrix} \vec{f} \\ \vec{m} \end{bmatrix}. \tag{4.11}$$

These are related by

$$\vec{\omega} = \mathbf{K}\vec{T} \tag{4.12}$$

and

$$\vec{T} = \mathbf{C}\vec{\omega} \tag{4.13}$$

where $\mathbf{K}$ and $\mathbf{C}$ are the 6x6 stiffness and compliance matrices, respectively.

For compliant mechanisms, the eigentwist was used as a representation of the mechanism's preferred motion. For a rotational mechanism, the eigentwist became the representation of a mechanism's preferred axis of rotation. This allowed the kinematics of the mechanism to be specifically characterized and compared to the kinematic requirements. Similarly to the compliance ellipsoid in Section 3.2, the eigentwist was derived from the 6x6 compliance matrix of a mechanism represented in

$$\vec{u} = \mathbf{C}\vec{f} \tag{4.14}$$

which is equivalent to Eq. 4.13 and can be expanded to

$$\begin{bmatrix} u_x \\ u_y \\ u_z \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} C_A & C_B \\ C_B & C_D \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{bmatrix} \tag{4.15}$$

where C_A, C_B, and C_D are all 3x3 matrices. The eigentwist was derived from the eigenvectors of C_D of the compliance matrix. The eigentwist provided an additional piece of information that the ellipsoid evaluation did not. Both produced the same rotation axis direction for a mechanism, but the eigentwist additionally placed that rotation axis in space. The two components of the eigentwist, δ and γ, were defined

using

$$\delta_1 = \frac{1}{\lambda_1}\mathbf{C}_B\gamma_1 \tag{4.16}$$

$$\delta_2 = \frac{1}{\lambda_2}\mathbf{C}_B\gamma_2 \tag{4.17}$$

$$\delta_3 = \frac{1}{\lambda_3}\mathbf{C}_B\gamma_3 \tag{4.18}$$

where the λ's are the eigenvalues and the γ's are the corresponding eigenvectors of C_D, and C_B is the lower, left 3x3 of the compliance matrix. This inclusion of C_B accounted for the coupling between rotation and translation, providing a complete description of the mechanism's preferred motion. It was stated previously that every rigid body displacement can be described by a rotation about an axis and translation along that axis, in which the rotation and translation are coupled. This was seen in Eqs. 4.16 to 4.18 where the rotation, γ, causes translation by being multiplied by the C_B. Considering this, each eigentwist was defined as 6x1 vector,

$$\vec{T} = \begin{bmatrix} \vec{\delta} \\ \\ \vec{\gamma} \end{bmatrix} \tag{4.19}$$

where $\vec{\delta}$ and $\vec{\gamma}$ are the translation and rotational components, respectively.

A simplified explanation of the eigentwist decomposition into screw parameters derived from work of Hopkins [41] and Davidson [42] is presented by Nijssen in [43]. The eigentwist was converted to the screw parameters ($\vec{q}$, $\vec{\omega}$, h, and θ) and compared to the requirement screws using

$$\vec{T} = \begin{bmatrix} \vec{\delta} \\ \\ \vec{\gamma} \end{bmatrix} = \begin{bmatrix} \vec{\omega} \times \vec{q} + h\vec{\omega} \\ \\ \vec{\omega} \end{bmatrix}. \tag{4.20}$$

Recalling that $\vec{\omega}$ is the rotation component of the screw, and with the knowledge that a rotational velocity is some direction causes a transition in that same direction, it was said that $\vec{\omega}$ was equal to $\vec{\gamma}$. Furthermore, it is generally know that a spatial translation is characterized by a pure translational component and a component caused by rotation. In the relationship described by Eq. 4.20, $h\vec{\omega}$ is the pure translational component, and $\vec{\omega} \times \vec{q}$ is the translational component caused by rotation.

Now that the reasoning of why screw parameters can be derived from an eigentwist has been presented, the calculation of each parameter is presented. The pitch, h, was found using

$$h = \frac{\vec{\omega} \cdot \vec{\delta}}{\vec{\omega} \cdot \vec{\omega}} \tag{4.21}$$

Now, h, δ, and $\vec{\omega}$ were known such that $\vec{\delta} = \vec{\omega} \times \vec{q} + h\vec{\omega}$ could be separated by components to solve for the components of the locating vector, $\vec{q}$,

$$\delta_x = q_y\omega_z - q_z\omega_y + h\omega_x \tag{4.22}$$

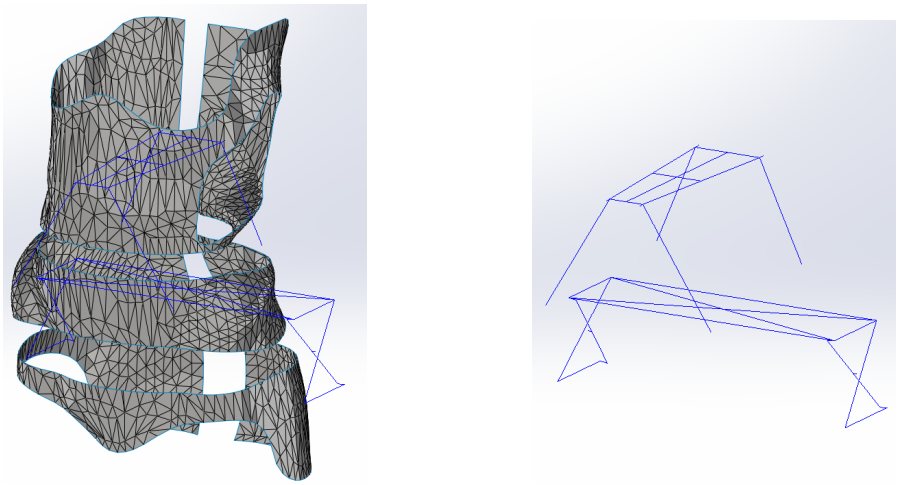
$$\delta_y = q_z\omega_x - q_x\omega_z + h\omega_y \tag{4.23}$$

$$\delta_z = q_x\omega_y - q_y\omega_x + h\omega_z \tag{4.24}$$

The eigentwist has now been defined using screw parameters, positioned with respect to the actuation point. The use of this decomposition to align a compliant mechanism's primary axis of rotation provides another large contribution to the field of compliant mechanisms, as the eigentwist has not previously been used to align mechanisms for a specific motion.

The eigentwist is now used to locate the primary axis of rotation of the designed mechanisms. A general representation of the mechanisms, designed to connect the brace pieces, are seen in Figure 4.9b. A different compliant joint was utilized for

lateral bending rather than a cartwheel hinge because the requirement screw for lateral bending is above the mechanism opening, near the middle of the upper brace piece. Therefore a cartwheel hinge, with axis of rotation at the intersection of the beams, was not capable of achieving the desired motion. Recognizing that the intersection of the beams in the cartwheel defines the axis of rotation, it was discerned that a triangle, or trapezoid, arrangement of beams itself would consist of similar compliant properties. Therefore, the nonparallel 2-beam mechanism was devised with the knowledge that the axis of rotation would be near the projected intersect of the two beams.

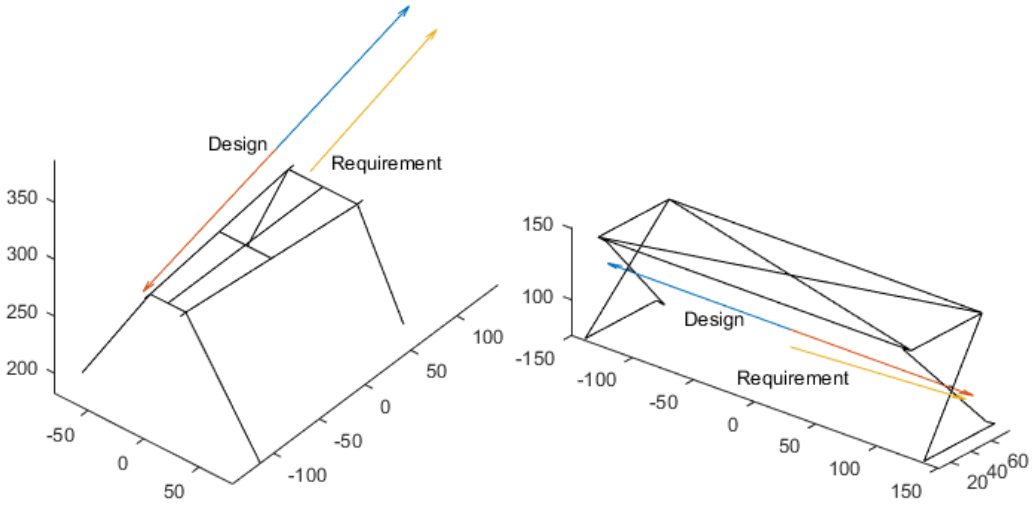


(a) **(b)**

Figure 4.9: Representations of the mechanisms used in FEA (a) in relation to brace and (b) pictured alone for a better visual.

In FEA, the lower 6 points on the cross-pivot and lower 4 on the two-beam mechanism were constrained in all directions. All beams other than the mechanism beams were modeled with large thicknesses to produce a very high stiffness compared to the flexible members. The middle point on the top of each mechanism was used for actuation. Knowing that the primary screw axis can be easily defined allowed for the adjustments to be made to align the screw as close as possible to the requirement screw. Figure 4.10 shows the eigentwist is positioned as predicted for the lateral

bending mechanism, along with depicting the screw of the cross-pivot mechanisms. The cross-pivot and two-beam mechanism were defined by the variables in Tables 4.3 and 4.4, respectively. The two thicknesses have no affect on the screw axis location because they do not affect the intersection of the flexure beam elements.



(a) **(b)**

Figure 4.10: The primary screws of motion for the chosen mechanisms in their final configurations.

Table 4.3: Parameters used for cartwheel hinge design.

Parameter	**Symbol**
Width	w
Height	h
Vertical Sagittal Location	z
Horizontal Sagittal Location	y
Angle Towards Torso	θ
Beam Thickness	t_{beam}
Planar Thickness	t_{planar}

Table 4.4: Parameters used for two-beam mechanism design.

Parameter	**Symbol**
Width (top front, bottom front, top back, bottom back)	$w_{tf}, w_{bf}, w_{tb}, w_{bb}$
Height (front, back)	h_f, h_b
Vertical Frontal Location (front, back)	z_f, z_b
Horizontal Frontal Location (front, back)	y_f, y_b
Angle Towards Torso (front, back)	θ_f, θ_b
Beam Thickness	t_{beam}
Planar Thickness	t_{planar}

Table 4.5 shows the screw parameters that describe the final design of the mechanisms, as seen in Figure 4.10. The screws do not exactly line up due to geometric constraints. The mechanism must attach to the brace pieces and therefore, the locations of the ends of the mechanisms are limited. As explained in Section 3.2.4, it is not completely necessary that the mechanisms model the prescribed screw of motion because all parts of the spine perform some portion of the motion. The ideal screw location is where the motion primarily occurs, but not the only place. Therefore, the errors noted in the table are consider allowable. The 5 degrees of error in flexion and 31 mm of error in lateral bending were caused by design assumptions. The axis for flexion was assumed to be exactly lateral, while the actual axis that resulting from motion capture was 5 degrees off of lateral. For lateral bending, it was assumed that the axis should occur in the middle of the torso, while in actuality, it occurred about 31 mm to one side. These assumptions were used to determine allowable error in mechanism design such that mechanism should be within 5 degrees of the desired direction of the rotation axis and within 31 mm of the desired location of the axis. The differences listed in Table 4.5 are all within this defined allowable error.

Table 4.5: List of requirements derived from motion capture.

	Requirement Screw	Design Screw	Difference
Flexion	$\vec{\omega} = [1, 0, 0]$	$\vec{\omega} = [0.99, -0.04, 0.08]$	5 deg
	$\vec{q} = [0, 44, 113]$	$\vec{q} = [0, 44, 100]$	[0, 0, 13] mm
Lateral Bending	$\vec{\omega} = [-.03, 0.95, 0.24]$	$\vec{\omega} = [0, 0.97, 0.25]$	1.7 deg
	$\vec{q} = [31, 0, 345]$	$\vec{q} = [-.01, 0, 349]$	[31, 0, -4] mm

4.3.2 Material Selection

Other than geometry, the material used to produce a mechanism causes one of the greatest effects on a mechanism's ability to perform a desired motion. To be successful, mechanisms must be flexible such that they can be displaced and strong such that they do not fail. In general, the ratio of yield strength to Young's modulus, S_y/E, is used as a primary factor when choosing material because it provides insight into how strong the material is with respect to its flexibility [14, 17]. A higher value is better. Even more important than yield strength is the fatigue strength of the metal. Assuming a patient performs 100 cycles per day, for 2 years, they actuate the mechanism 73,000 cycle, so the fatigue strength at 10^5 cycles was used for the ratio, $S_{fatigue}/E$. The Ashby Chart in Figure 4.11 was consulted to find ideal materials.

Delrin is known to be a good material for compliant mechanisms and has a relatively high ratio of 12, but it was found that Delrin does not have the strength needed for this application. Therefore, the option to use metal was explored. Though manufacturing is more difficult for metal, the higher strength makes it necessary. Phosphor bronze CUSN8, another material previously used in compliant mechanisms, was first chosen with a ratio of 3.64. This however was also not strong enough. Titanium alloys reach farthest to the right, the strength direction. Titanium Ti-10V-2Fe-3Al was found to have a ratio of 6.36. This metal was chosen and led to a positive outcome.

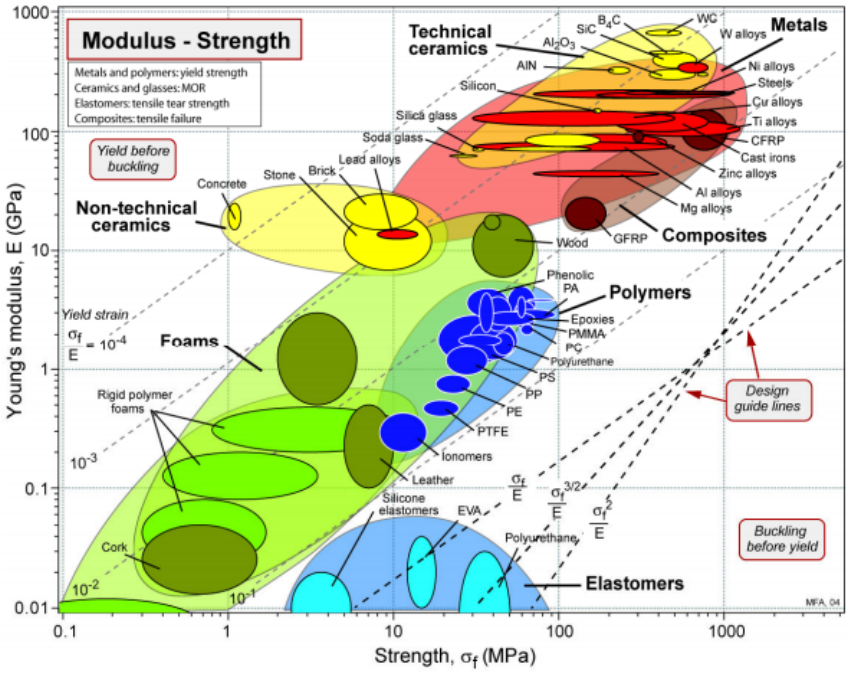


Figure 4.11: Ashby chart used for material selection [18].

Table 4.6 contains general properties for these materials.

Table 4.6: Material Properties

Material	E (GPa)	Sy (MPa)	Sf (MPa)	(Sy/E) x 1000	(Sf/E) x 1000	Density (kg/m3)	Poisson
Delrin	2.95	69	35	23.4	11.86	1420	0.35
CUSN8	110	700	400	6.4	3.64	8780	0.35
Titanium	110	1240	700	11	6.36	4650	0.33

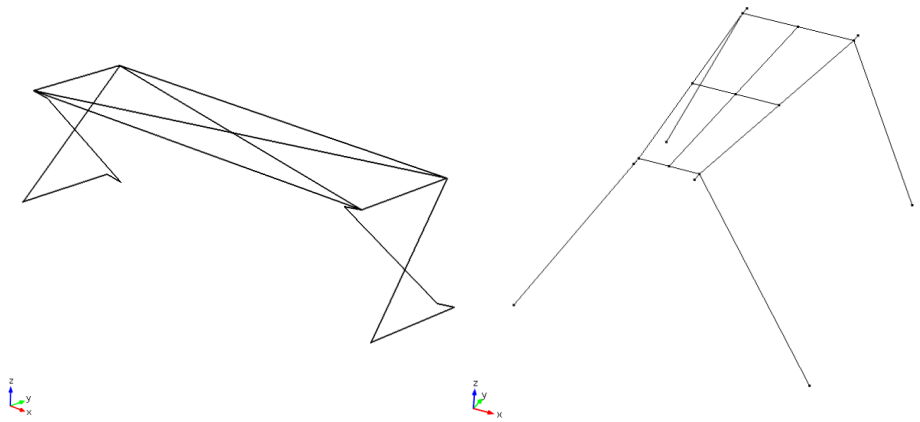
4.3.3 Beam Thickness

The same FE models were used to define the optimal beam thickness. The models used in design can be seen in Figure 4.12 without the screws. For this analysis, geometric nonlinearities were included and COMSOL's *Automatic highly nonlinear (Newton)* method was used. The Euler-Bernoulli beam model was used in this anal-

ysis because the beams were long and slender. In general, these types of flexures are expected to have low strain, so the Euler-Bernoulli beam can be used. The strain was estimated to be 0.032 for the longer of the upper mechanism beams using

$$\epsilon_{max} = \frac{My}{IE} = \frac{Fl\frac{h}{2}}{\frac{1}{12}bh^3E} \tag{4.25}$$

for a cantilever beam. F is the force on the end of beam, l is the length of the beam, h is the thickness of the beam, b is the width of the beam, and E is the Young's modulus. The estimated strain is small enough to suggest that the Euler-Bernoulli beam is a good model for the flexures.



(a) Cross-Pivot **(b)** Two-Beam Mechanisms

Figure 4.12: Mechanisms as they were analyzed in COMSOL.

To determine the ideal beam thickness, a parametric study was completed. The required force system, containing three forces and three moments, (Table 4.7) were applied to the actuation point with the addition of up to a 50N actuation force such that the patient could provide. This actuation force was estimated by attaching a spring scale to a subject through use of a belt and having the subject perform a motion (Figure 4.13). This magnitude of force was relatively easy to reach. The difference

between the resulting rotation angle at 0N of actuation and the angle at 50N of actuation defines the patient's mobility and must be compared to the requirements also noted in Table 4.7.

Table 4.7: Force Requirements

	Angle	**Resultant Force (N)**			**Resultant Moment (N-mm)**		
Location		x	y	z	x	y	z
Upper-Middle	7.28^o	19	37	8.6	-0.9	0.3	0.4
Middle-Lower	30.8^o	-22	18	-2.5	-1.2	0.7	0.6

Figure 4.13: Depiction of test used to estimate the force that the torso can apply to the brace. The red arrow denotes the direction of motion.

It was determined that the mechanisms can achieve the most rotational motion when the beams are thickest without reaching their fatigue strength when the 50N load is applied. The method used to determine this criterion is explained in Appendix B. As stated in Section 4.3.2, Titanium Ti-10V-2Fe-3Al was chosen to be the ideal material. The optimal thicknesses, 1.2 mm and 0.78 mm, for flexion and lateral bending mechanisms, respectively, can be seen in Tables 4.8 and 4.9. The overall final dimensions for the beams are noted in Table 4.10.

Table 4.8: Optimal beam thickness for cartwheel hinge.

Beam Width (mm)	**Force (N)**	**Angle (°)**	**Stress (MPa)**	**Angle Diff(°)**
1.2	0	8.7	195	
	50	35.7	694	27

Table 4.9: Optimal beam thickness for two-beam mechanism.

Beam Width (mm)	Force (N)	Angle (°)	Stress (MPa)	Angle Diff(°)
0.78	0	1.17	205	
	50	6.65	706	5.48

These rotation angles do not quite reach the goal set for range of motion. The cross-pivots, which achieve 27°, were required to rotate 31°. The two-beam mechanisms, which achieve 5.48°, were required to rotate 7.28°. That is 88% and 75% of the original goal for the cross-pivot and two-beam, respectively.

To achieve the pre-displacement described in Section 3.3.1, the mounting points of the mechanisms were rotated by the zero-force angles (8.7^o and 1.17^o) noted in Tables 4.8 and 4.9 with respect to the designed rotation axis of each mechanism. Again, the zero-force angle is the displacement of the mechanisms when corrective forces are applied, but no actuation force is applied.

Table 4.10: Overall dimensions for all beams.

Beam	Width (mm)	Thickness (mm)	Length (mm)
Flexion	6	1.2	103
Lat. Bend. Front	12	0.78	121
Lat. Bend. Back	12	0.78	96.2

4.4 Conclusion

At this point, the brace was fully designed. The expanded building block approach and the specialized eigentwist decomposition were developed and used to design for the motion requirements. These two tools, pertaining to the field of compliant mechanisms, provided the main contribution of this chapter. Where as previously developed

synthesis methods were limited to planar problems or a particular type of compliant mechanism, the expanded spatial building block approach can be utilized with any type of compliant mechanisms in both planar and spatial problems. Pertaining to the eigentwist, a new use for it was developed such that the exact rotation axis for the linear range of a mechanism can be identified and aligned with a requirement. Both of these tool are can be considered very general such that they can be used in most, if not all, compliant mechanism design problems.

Chapter 5

The Scoliosis Brace and Testing

Two tests were performed to determine the value of the design: testing the efficacy of the brace and confirming the FE model prediction. The efficacy test involves modeling the brace in a way such that it can be applied to the FE torso developed at École Polytechnique de Montréal. Their model is proven to correctly predict the correction that a brace will apply to the spine. The second test involved applying a displacement to the brace and measuring the applied load using a load cell to produce a force-displacement curve.

5.1 Constructing the Physical Brace

The brace was manufactured using rapid prototype and machining techniques. Three types of parts were used in the assembly: brace pieces, mechanism beams, and mechanism connectors. The mechanism connectors are the link between the brace pieces and the mechanism beams. The brace pieces were 3D printed out of ABS with a

thickness of 4 mm. The mechanism beams were machined from titanium alloy sheet. The brace connections were also 3D printed. An assembled CAD model of the brace can be seen in Figure 5.1.

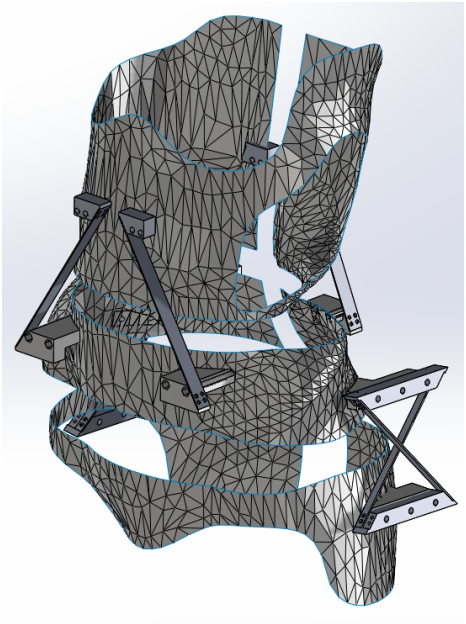
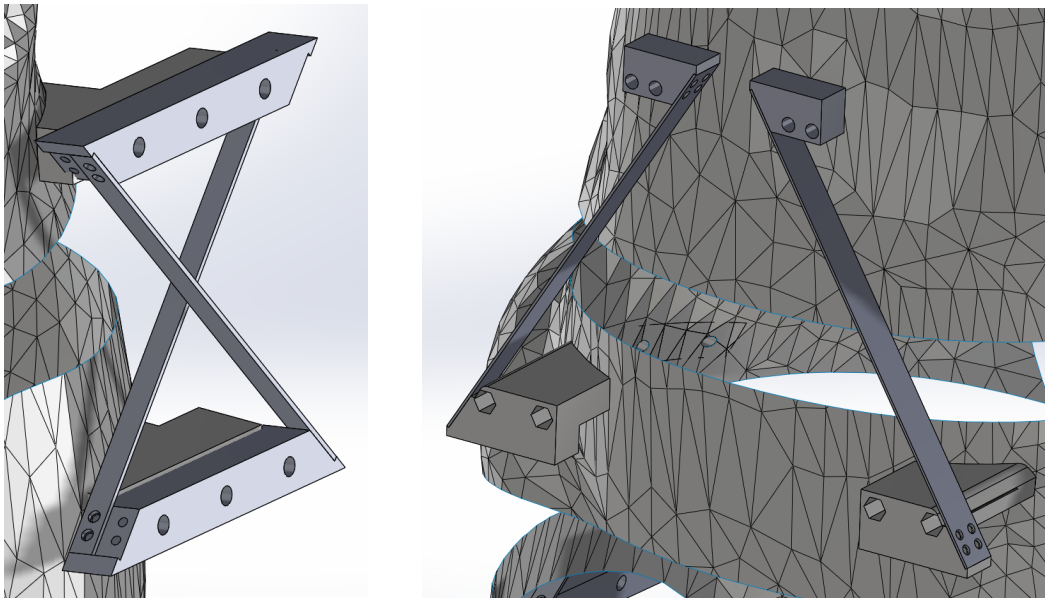


Figure 5.1: Brace Assembly.

Close-up views of the mechanism assemblies are seen in Figure 5.2. For the prototype, the beams are bolted to the connections and the mechanism connections are bolted to the brace pieces. The connections are stiff such that they contribute a minimal amount of compliance compared to the beams. For the cross-pivot, a 1 mm space was designed between the two beams, preventing contact between the them. This is not a final design for production, but a first proof-of-concept prototype.



(a) Cross-Pivot **(b)** Two-Beam Mechanism

Figure 5.2: Detailed images of mechanism assemblies.

5.1.1 Manufacturing Errors

The small thickness of the beams caused difficulty in manufacturing. To secure the beams for thickness milling, they were glued to a steel plate. As a result, the thicknesses were not exactly as prescribed, and the beams for the two-beam mechanisms became slightly bowed during machining (Figure 5.3). The actual thickness, 1.19 mm, for the cross-pivot beams is slightly smaller than the 1.2 mm design. The thickness, 0.787 mm, for the two-beam mechanism is slightly larger than the 0.78 mm prescribed design. In addition to the thickness error, the beams for the two-beam mechanisms were bowed. The longer beams (front mechanism) was bowed 1.78 mm measured at the center, while the shorter beams (back mechanism) was bowed 1.02 mm at the center.

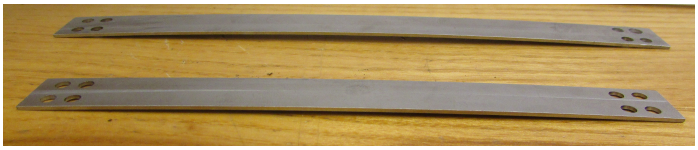


Figure 5.3: Image showing the bowing caused in manufacturing of the beams.

5.2 FEA Confirmation

5.2.1 Test Assembly

An assembly (5.4) was created for the stiffness testing of the mechanisms in order to create accurate force-displacement curves. These curves are to be compared to curves predicted by the FE model. It was essential to ensure that the structure between the mechanism beams was stiff because the finite element model assumes this. The bottom of the cross-pivots and the tops of the two-beams mechanisms were connected by aluminum bars. This also allowed for easy mounting to an aluminum plate at the bottom and load cell at the top. To create stiffness in the middle brace piece, it was printed with 10 mm thickness, and a plate was used to create a rigid connection at the opening of the brace piece.

An ATI Mini45 6-axis load cell is mounted between the top bar and a handle. On top of the handle is a plate with Vicon markers used to capture the motion of the brace. To remove error caused by the brace itself moving within the reference frame, both the brace and reference frame were mounted to the same 1/4" thick plate.

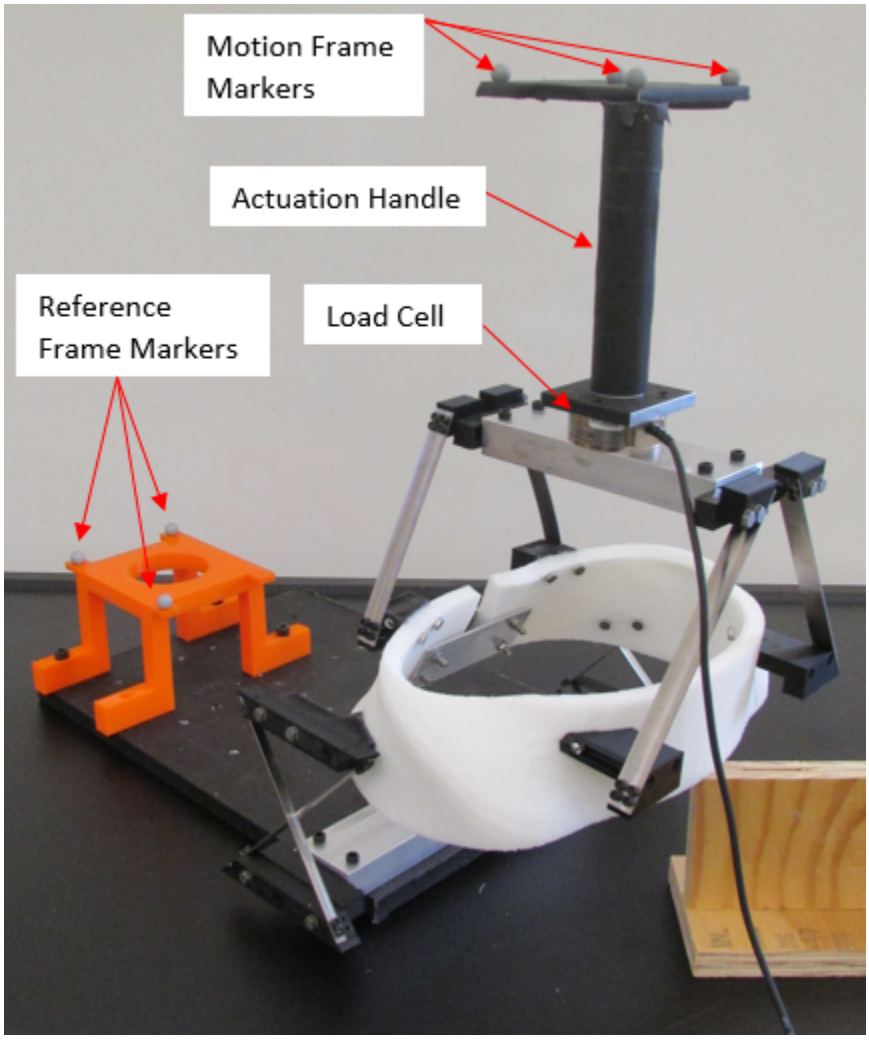


Figure 5.4: Test setup.

5.2.2 Data Collection

The test data was taken in a very similar manner to the motion capture performed for the kinematic characterization. The motion of a set of 3 markers was captured though actuation. Force/moment data was also simultaneously recorded at 1000 Hz from the load cell, connected through the Vicon MX data acquisition unit. The motion data was analyzed using the screw analysis once again, with the main results being the

the angle of motion in response to a force input. The screw analysis was performed relative to this reference frame, assuming the zero brace frame to be the actuation handle in the vertical position.

5.2.3 Force - Displacement Curve Verification

To compare the FE modeled brace to the physical prototype, the FE model was actuated by applying the forces recorded through a motion from the physical test, with the output being a displacement. Before running the FE model, the beam thicknesses were corrected for the actual machined thicknesses. Since the finite element program applies forces in a constant direction through the motion and the load cell's orientation changes through a motion, the force components were corrected before being applied to the FE model. The load cell data was filtered using MATLAB's *Savitzky-Golay filtering* function because it was noisy. Also, the load cell recorded data at 10x the frequency of the motion capture, so a lower pass filter using MATLAB's *decimate* function was applied to reduce the load cell data to match the Vicon's 100 Hz data.

The model was run for the two primary motions, flexion and lateral bending. All other directions are stiff and allow minimal motion. In the physical testing, the handle was pushed forward in the sagittal plane to achieve flexion for the cross-pivot mechanisms and a moment was applied for lateral bending. The force-displacement curves for flexion and lateral bending can be seen in Figures 5.5 and 5.6, respectively. For these plots, the forces and moments from *Physical Test 1* were applied in the FE model to determine the resulting rotation magnitude for comparison with the physical test.

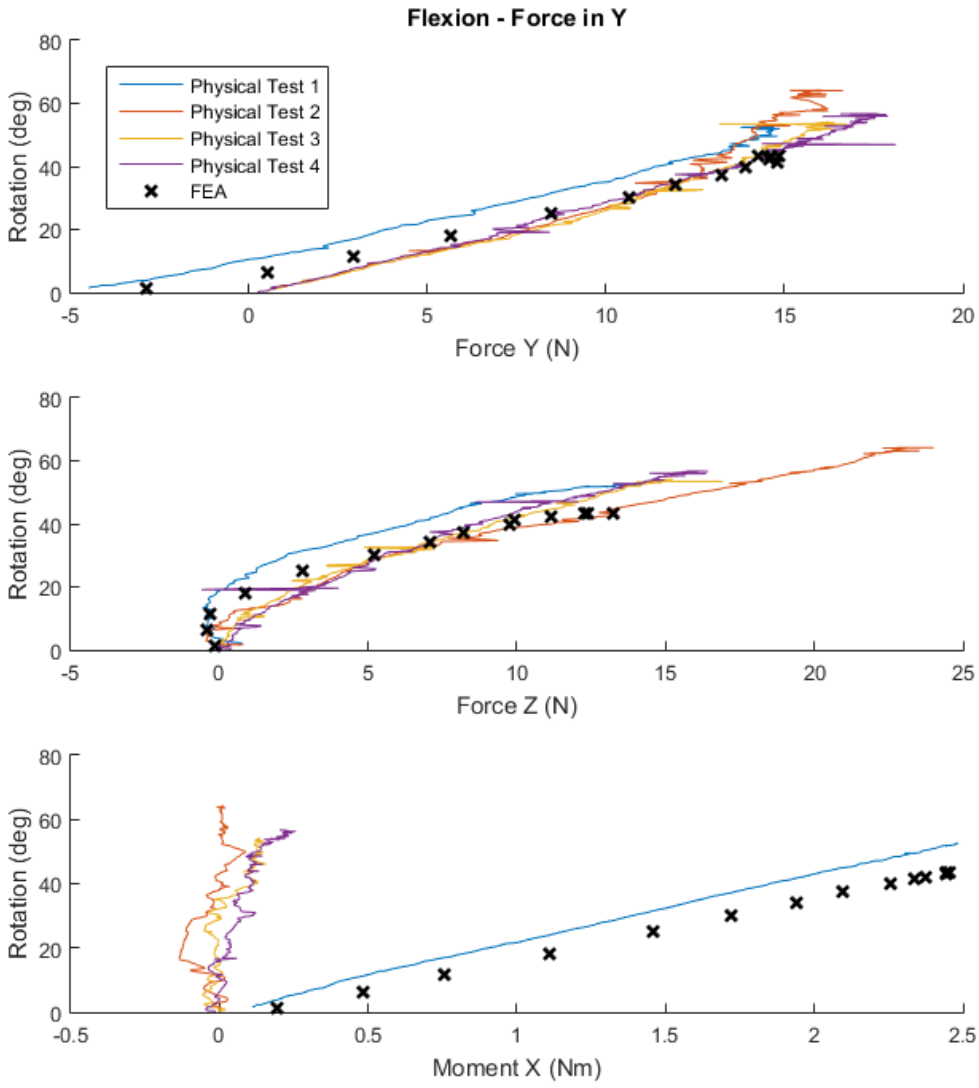


Figure 5.5: Force-displacement curves for flexion mechanisms.

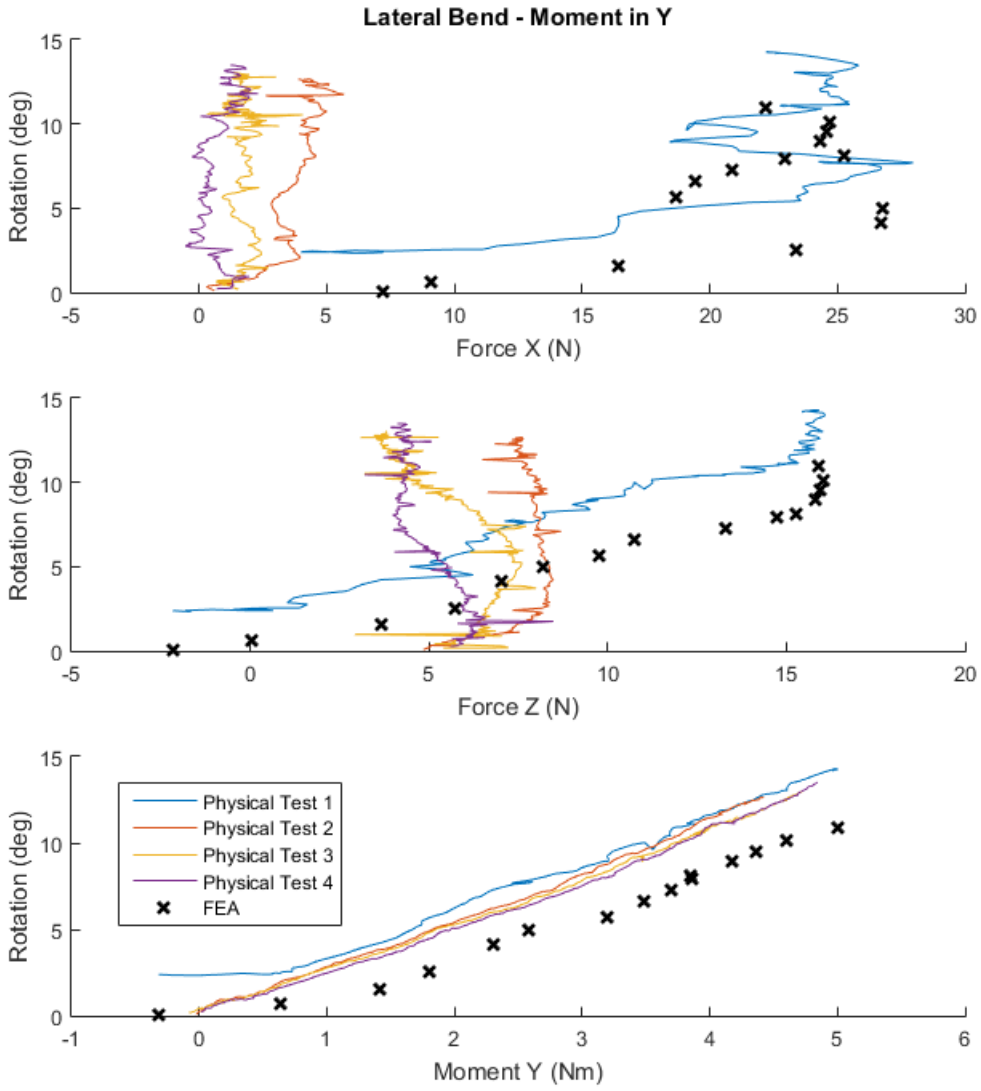


Figure 5.6: Force-displacement curves for lateral bending mechanisms.

Through inspection of the plots, it is seen that the moment for flexion and the forces for lateral bending are very inconsistent. This is easily explained recalling the type of force used to actuate each mechanism and where the actuation occurred. For flexion, a force was applied at a significant distance from the designed rotation axis.

The force actuation was chosen for this case because of the distance. A force applied at a distance from a rotation axis causes a significant deformation. Conversely, a moment applied at a distance from an axis minimally affects the deformation of the mechanism. For this reason, the important data examine for flexion is the forces. The opposite is true for the lateral bending mechanisms, for which a moment was applied near the axis of rotation. Therefore, the moment data was inspected to determine the correlation between the FEA and the physical test.

The trends of the FE model and Physical Test 1 match for both motions. The error in magnitudes was calculated using

$$R^2 = 1 - \frac{\sum \left(F_{actual} - F_{FEA}\right)^2}{\sum \left(F_{actual} - \bar{F}_{actual}\right)^2} \tag{5.1}$$

and can be seen in Figure 5.1. For the flexion data, the two force plots show that the actual mechanism behaves similarly to the FE model. is more compliant than the FE model, because the FE model has greater forces at the same displacement angles. As per the R^2 error, only the force in the z-direction has a questionable correlation between the FE model and Physical Test 1, but other three test are more highly correlated.

The moment for the lateral mechanisms suggests that the actual mechanism is more compliant than the FE model. The trend of the FE model and the physical tests match closely because both have a linearly increasing rotation as a greater moment was applied.

Table 5.1: Maximum errors in FEA when compared to Physical Test 1.

	Data	$\mathbf{R^2}$
Flexion	Force Y	0.90
	Force Z	0.39
	Moment X	0.82
Lateral Bending	Force X	0.73
	Force Z	0.95
	Moment Y	0.96

5.2.4 Sources of Error

This error could stem from a few different causes. The flexion motion is highly nonlinear, and COMSOL may not be the ideal nonlinear solver for this problem. Variation in beam thickness could also contribute to error. Overall, compliance in the 3D-printed parts could easily reduce the stiffness of the overall system. This would definitely explain the variation between the physical model and the FEA prediction.

This error of lower stiffness would reduce the brace's ability to correct the spine because the brace will apply smaller forces causing less displacement of the spine. Even if only one mechanism provides lower stiffness, the displacement of each brace piece depends on the equilibrium between all brace pieces so the overall brace stiffness will be reduced. Even just one incorrect mechanism can cause the whole brace to fail to be unsuccessful in correcting the spine.

5.3 Efficacy Test

Collaborators at École Polytechnique de Montréal have a system that models a specific patient's torso such that a model of a brace can be applied and tested for its ability to be successful. A full description of the model can be found in [24]. This model

has been proven successful in determining the efficacy of a standard rigid brace. A sample result for one of their rigid braces was used to derive the requirement forces for this project. It is considered successfully brace and the results of their simulation can be seen in Figure 5.7. It can be seen that both apexes of this S-curve have been reduced.

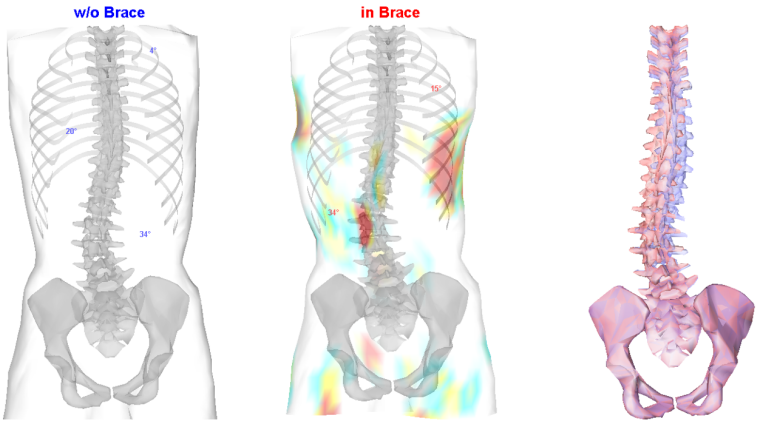
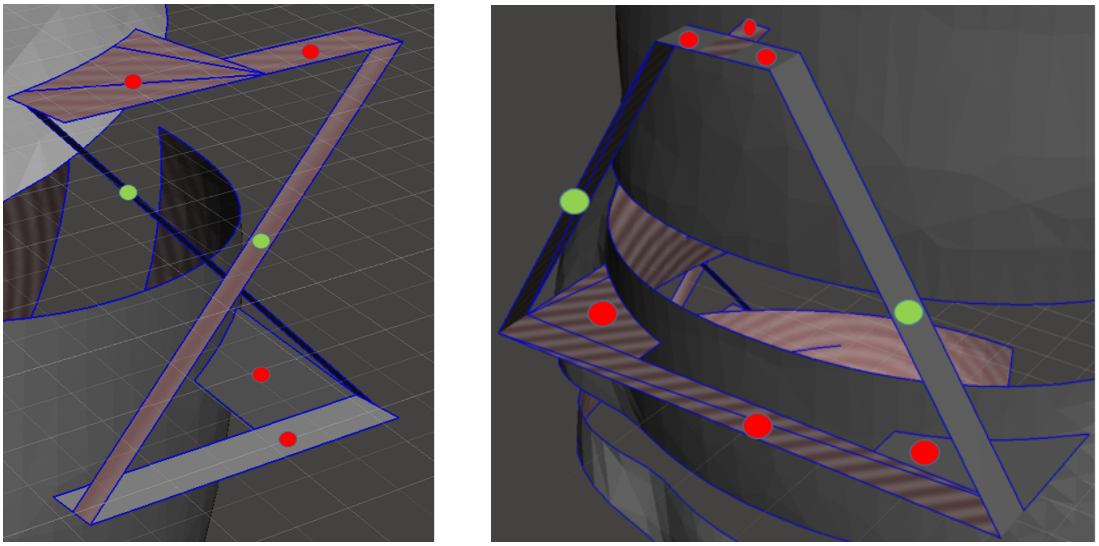


Figure 5.7: (left) Pressure applied to torso by the original rigid brace, and (right) comparison of spine in the brace (blue) and without the brace (red).

Their system had to be modified to accept a brace made of multiple pieces, such that the compliant brace could be tested. Originally, a model of the brace with unoptimized beams was sent to our collaborators to see if it was possible to test the compliant brace on their system. Since they primarily used shells as elements in their model, the compliant brace and mechanisms were built from shells. The mechanisms, as modeled can be seen in Figure 5.8, and the brace pieces can be seen in Figure 5.9. They were able to modify their system and perform a successful simulation. It as found that this unoptimized brace would be unsuccessful because it was not stiff enough, causing no correction. These results can be seen in Figure 5.10.



(a) Cross-Pivot **(b)** Two-Beam Mechanism

Figure 5.8: Mechanisms as modeled for efficacy test. Red and green dots mark rigid and flexible members, respectively.

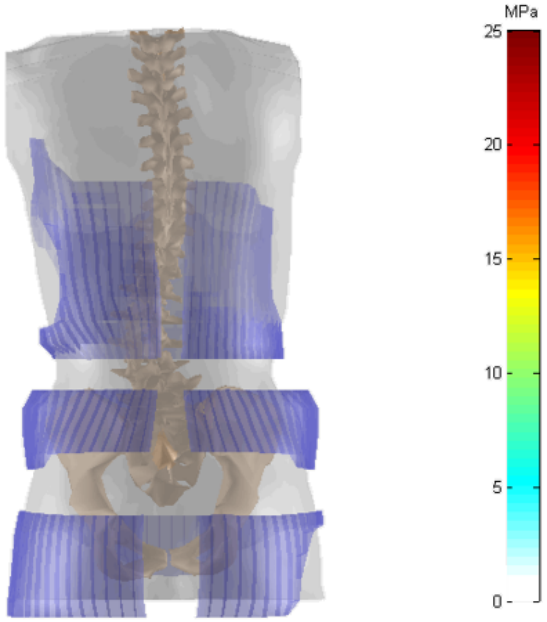


Figure 5.9: Brace pieces as positions on torso for efficacy test.

After the brace design was finished and optimized, the updated model was sent to them and tested in their system. The results were minor correction, but also higher

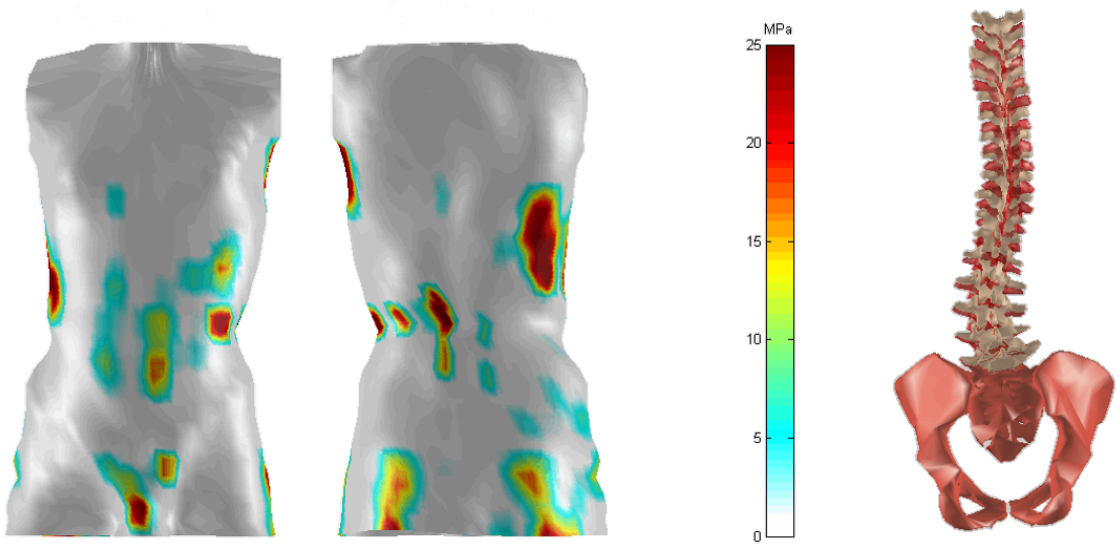


Figure 5.10: (left) Pressure applied torso by initial test flexure brace, and (right) comparison of spine in the brace (red) and without the brace (white).

pressures. It can be seen that the general locations of high pressure are in the same regions at the original brace. This proves that the brace is applying force in the correct locations. It should be noted, though, that the compliant design applied the large forces on smaller areas, causing high pressures. This specifically occurs in the mid-lower back region, as noted in Figure 5.11. The extra pressure is likely caused by the removal of material, as it can be seen in Figure 5.9 that material was removed in this location leaving less area to distribute the force.

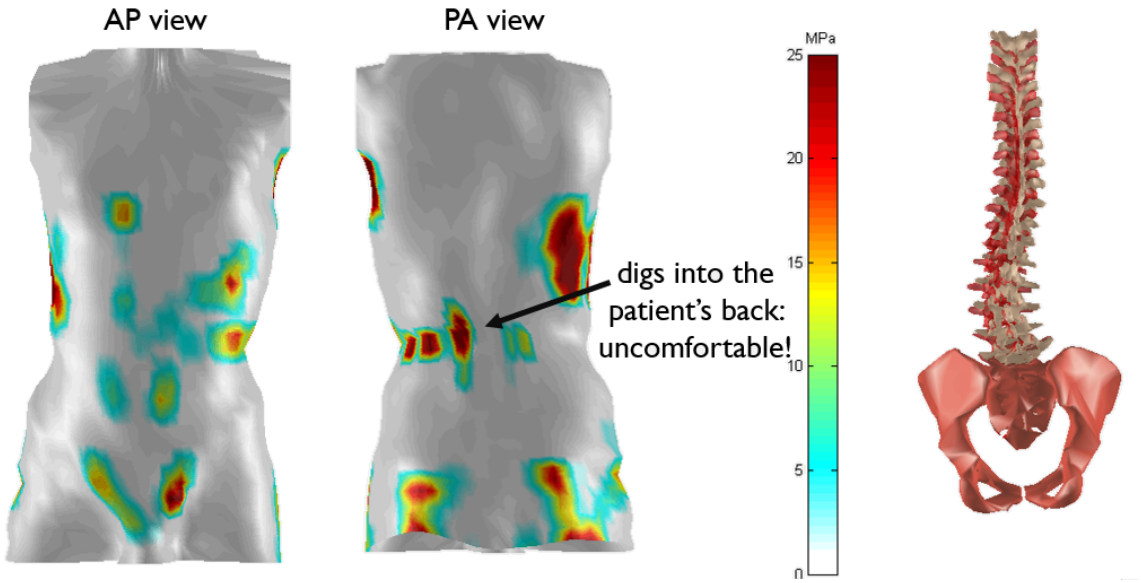


Figure 5.11: (left) Pressure applied torso by final flexure brace, and (right) comparison of spine in the brace (red) and without the brace (white).

Although the results are not completely positive, they are promising. The brace does cause some correction. This could be improved if the required force system is better understood. The high pressures were most likely caused by the point load assumption. The location of forces was known to the body, but had to be estimated in location on the brace. In the future, it would be best to get a report of the pressures mapped to the brace rather than the torso. This would provide an exact description of where on the brace pressure is applied, better informing where material can be removed. Then, the brace structure could be removed where there is little or no force, allowing for better force distribution.

Chapter 6

Conclusions & Future Work

This thesis provided contributions to the fields of compliant mechanisms, spinal motion characterization, and scoliosis bracing with the general contribution of developing a new scoliosis brace. Each of the contributions offers opportunity to be furthered to produce improved and more general results. This chapter discusses these contributions and future work.

6.1 Key Contributions

The *3-dimensional compliance ellipsoid characterization of mechanisms* was the first major contribution of this thesis. It pertains to the field of compliant mechanism design, further expanding the Building Block Approach initially developed for 2D synthesis. With this expansion, any compliant mechanism can be characterized in 3D through inspection of the mechanism's compliance matrix allowing for the creation of a library of 3D building blocks. These building blocks can be visually compared

to design requirements characterized as ellipsoids with the goal of choosing possible mechanisms. The use of the decomposed eigentwist used to align the mechanism's rotation axis was introduced in this thesis. This use of the eigentwist has not previously been explored in literature.

The scoliosis brace was a *new use for compliant mechanisms*, and it is notable that mechanisms can follow the body's natural motion. This breaks the ground for using compliant mechanism for other problems that relate to the body's motion, such as prosthesis or other bracing necessities.

Two contributions were made in the way of characterizing spinal motion. The *screw theory analysis* allows researchers to find exact axes of rotation for motion. As seen with the hip motion characterization, the screw theory analysis can also be applied to other parts of the body. The *piece-wise description of the spine motion* is also noteworthy. Rather than just describing the the motion with respect to the lumbar or thoracic regions, these regions were separated into smaller subregions such that each subregion of the spine was described. This led to finding where in the spine each of the three primary motions occur.

Lastly, to the field of scoliosis bracing, the *force design paradigm* was explored. Rather than correcting with displacement as the only goal, designing for compliant mechanisms required considering both force and displacement. In order to design within this paradigm, the fully defined correctional force system was determined, providing key insights into how much force is actually imparted on the body. It was found that knowledge of the forces on the torso can be gained through integrating a pressure map of the brace on the torso. This pressure map had not previously been used to determine force maginitudes.

6.2 Discussion & Future Work

A good foundation of tools has been developed to be able to further iterate and design the scoliosis brace. Most of these tools have room for improvement. With respect to the motion characterization, a study with many subjects is necessary to make more general statements about the motion of the torso. If this is done, a general brace design could be used for all patients with slight adjustments. Currently, only a single subject's motion was characterized. The process used to perform the motion capture can also be improved. Currently, three markers were placed on the skin and were assumed to stay in the same position relative to each other. If a device that constituted a rigid body could be placed directly on the spine, then this assumption could be eliminated and error could be reduced. Lastly, the piecewise description of the spine was separated into four sections. If markers can be placed even closer together, the spinal motion could be separated into more sections, and an even better piecewise description of the motion could be produced.

It was mentioned that kinematic error is allowable because the spine can perform a motion in many different ways. Determining the amount of allowable error will help better define the kinematic requirements. One test that could be performed is constraining portions of the spine and determining to what capacity a patient can still perform a motion. With regard to the kinematics, the assumption of averaging the screws of motion to a single design requirement could be omitted. Instead, the movement of the screws in space as the patient performs a motion can be considered. Connecting the movement of kinematic screws to design, a nonlinear eigentwist analysis of the mechanisms could be used to model the drift of the mechanism's rotation axis through motion. The drift of the mechanism's motion axis can then be designed to follow the movement of the kinematic screws.

Further progress is also required in the way of finding the forces applied to the body by a brace. A pressure map could be gained from a model as was done in this thesis, or it could be determine by applying a pressure mapping sensor between a patient and their brace. A study with many patients will give a better idea of the necessary forces needed to cause corrective displacement. An ultimate goal would be to draw a correlation between a patient's Cobb angle and the force required for correction.

The brace itself is far from a final product. Nijenbanning emphasized that in addition to the brace's function, it must be cosmetically appealing. One necessary next step is to explore possibilities of bringing mechanisms closer to the torso such that they are less noticeable and more easily hidden. This could lead to a new direction with respect to exploring compliant mechanisms. The planar topology of the flexures could be projected on the curved surface of the standard brace. Investigating their behavior in these odd shapes would provide an interesting contribution to the field of compliant mechanisms and also allow for a better brace.

Practically, it was found that the specific brace design would not provide enough correction to prevent further progression of the scoliosis curve. Rather than just iterating again, it is important to close the loop and understand why the brace did not work as designed. The pressure maps from the efficacy test for the compliant brace should be compared to original pressure map from the successful brace. This comparison can inform how the location and magnitudes of forces vary between the two and inform how the compliant brace might be modified to achieve the necessary correction. In addition, the brace did not allow the required magnitude of motion. The primary limitation was the material's fatigue strength. In general, a lower fatigue strength is arises from a mechanism to performing more cycles. One path to explore

to increase range of motion is to assume that the mechanism can be replaced after some period of time such that the mechanism needs to perform fewer cycles and therefore has a higher fatigue strength. The mechanisms will be able to perform a great range of motion before reach the higher fatigue strength.

6.3 Conclusion

The overall outcome of this thesis was the development of a scoliosis brace. This brace has the potential to allow the patient greater mobility. The expanded Building Block Approach was successful in allowing for the decomposition of the design problem and generation of possible solutions. A material was found and the dimensions of the mechanisms were optimized to produce the required corrective force system. The FE model was confirmed with a physical test, and the efficacy was tested to be unsuccessful, although promising.

References

[1] Bible, J. S., Biswas, D., Miller, C. P., Whang, P. G., and Grauer, J. N., 2008. "Normal functional range of motion of the lumbar spine during 15 activities of daily living". *Journal of Spinal Disorders & Techniques,* **23**(2), pp. 106–112.

[2] Vissera, D., Xuea, D., Ronskya, J. L., Harderb, J., and Zernickea, R. F., 2012. "Computer-aided optimal design of custom scoliosis braces considering clinical and patient evaluations". *Computer Methods and Programs in Biomedicine,* **107**, pp. 478–489.

[3] MO: Mosby Elsevier, 2009. Mosby's medical dictionary.

[4] Fortin, D., Cheriet, F., Beausejour, M., Debanne, P., Joncas, J., and Labelle, H., 2007. "A 3d visualization tool for the design and customization of spinal braces". *Computerized Medical Imaging and Graphics,* **31**(8), pp. 614–624.

[5] Weiss, H. R., and Rigo, M., 2008. *The Conservative Scoliosis Treatment:1st SOSORT Instructional Course Lectures Book (Studies in Health Technology and Informatics).* IOS Press.

[6] Coillard, C., Vachon, V., Circo, A., Beausejour, M., and Rivard, C., 2007. "Effectiveness of the spinecor brace based on the new standardized criteria proposed by the scoliosis research society for adolescent idiopathic scoliosis". *Journal of Pediatric Orthopaedics,* **27**(4), pp. 375–379.

[7] Bulthuis, G. J., Veldhuizen, A. G., and Nijenbanning, G., 2008. "Clinical effect of continuous corrective force delivery in the non-operative treatment of idiopathic scoliosis: a prospective cohort study of the triac-brace". *European Spine Journal,* **17**, pp. 231–239.

[8] San Diego Center for Spinal Disorders, 2015. Explaining spinal anatomy. `http://www.sandiego-spine.com/subject.php?pn=spinal-anatomy-024`.

[9] Engsberg, J. R., Lenke, L. G., Uhrich, M. L., Ross, S. A., and Bridwell, K. H., 2003. "Prospective comparison of gait and trunk range of motion in adolescents

with idiopathic thoracic scoliosis undergoing anterior or posterior spinal fusion". *Spine,* **28**(17), pp. 1993–2000.

[10] Nijenbanning, G., 1998. "Scoliosis redress: Design of a force controlled orthosis". PhD thesis, Universiteit Twente.

[11] Midha, A., Bapat, S. G., Mavanthoor, A., and Chinta, V., 2015. "Analysis of a fixed-guided compliant beam with an inflection point using the pseudo-rigid-body model concept". *ASME. J. Mechanisms Robotics,* **7**(3).

[12] Hopkins, J. B., 2015. "A visualization approach for analyzing and synthesizing serial flexure elements". *Journal of Mechanisms and Robotics,* **7**(3).

[13] Liu, C.-H., and Huang, G.-F., 2016. "A topology optimization method with constant volume fraction during iterations for design of compliant mechanisms". *Journal of Mechanisms and Robotics,* **8**(4).

[14] Howell, L. L., 2001. *Compliant Mechanisms.* Wiley, New York.

[15] Baker, M. J., 2016. Euclidean space. `http://www.euclideanspace.com/maths/geometry/affine/screwTheory/`.

[16] Kim, C. J., 2005. "A conceptual approach to the computational synthesis of compliant mechanisms". PhD thesis, The University of Michigan.

[17] Howell, L. L., Magleby, S. P., and Olsen, B. M., eds., 2013. *Handbook of Compliant Mechanisms.* Wiley, West Sussex, United Kingdom.

[18] Granta Material Intelligence, 2010. Materials and process selection charts. `http://www.grantadesign.com/download/pdf/teaching_resource_books/2-Materials-Charts-2010.pdf`.

[19] Stokes, I. A. F., Bigalow, L. C., and Moreland, M. S., 1987. "Three-dimensional spinal curvature in idiopathic scoliosis". *Journal of Orthopaedic Research,* **5**(1), pp. 102–113.

[20] Weinstein, S. L., Dolan, L. A., Wright, J. G., and Dobbs, M. B., 2013. "Effects of bracing in adolescents". *The New England Journal of Medicine*, pp. 1512–1521.

[21] Nnadi, C., and Fairbank, J., 2010. "Scoliosis: a review". *Paediatrics and Child Health,* **20**(5), pp. 215–220.

[22] Asher, M. A., and Burton, D. C., 2006. "Adolescent idiopathic scoliosis: natural history and long term treatment effects". *Scoliosis.*

[23] Czaprowski, D., Kotwicki, T., Biernat, R., Urniaz, J., and Ronikier, A., 2012. "Physical capacity of girls with mild and moderate idiopathic scoliosis: influence of the size, length and number of curvatures". *European Spine Journal,* **21**(6), pp. 109–1105.

[24] Desbiens-Blais, F., Clin, J., Parent, S., Labelle, H., and Aubin, C.-E., 2012. "New brace design combining cad/cam and biomechanical simulation for the treatment of adolescent idiopathic scoliosis". *Clinical Biomechanics,* **27**(10), pp. 99–1005.

[25] Smith, S., 2000. *Elements of Elastic Mechanisms.* Taylor & Francis, London, England.

[26] Taylor, J. R., Liston, C. B., and Twomey, L. T., 1982. "Scoliosis: A review". *Australian Journal of Physiotherapy,* **28**(3), pp. 20–25.

[27] Rigo, M., and Weiss, H., 2008. "The cheneau concept of bracing - biomechanical aspects". *Studies in Health Technology and Informatics,* **135**, pp. 303–319.

[28] Wong, M. S., Cheng, J. C. Y., Lam, T. P., Ng, B. K. W., Sin, S. W., Lee-Shum, S. L. F., Chow, D. H. K., and Tam, S. Y. P., 2008. "The effect of rigid versus flexible spinal orthosis on the clinical efficacy and acceptance of the patients with adolescent idiopathic scoliosis". *Spine,* **33**(12), pp. 1360–1365.

[29] Guo, J., Lam, T. P., Wong, M. S., Ng, B. K. W., Lee, K. M., Liu, K. L., Hung, L. H., Lau, A. H. Y., Sin, S. W., Kwok, W. K., Yu, F. W. P., Qui, Y., and Cheng, J. C. Y., 2014. "A prospective randomized controlled study on the treatment outcome of spinecor brace versus rigid brace for adolescent idiopathic scoliosis with follow-up according to the srs standardized criteria". *European Spine Journal,* **23**(12), pp. 2650–2657.

[30] Mannion, A., and Troke, M., 1999. "A comparison of two motion analysis devices used in the measurement of lumbar spinal mobility". *Clinical Biomechanics,* **14**, pp. 612–619.

[31] Shum, G. L. K., Crosbie, J., and Lee, R. Y. W., 2007. "Movement coordination of the lumbar spine and hip during a picking up activity in low back pain subjects". *European Spine Journal,* **16**, pp. 749–758.

[32] BOSTON BRACE INTERNATIONAL, INC., 2003. *Reference Manual for the Boston Scoliosis Brace.*

[33] Clin, J., Aubin, C.-E., Parent, S., Sangole, A., and Labelle, H., 2010. "Comparison of the biomechanical 3d efficiency of different brace designs for the treatment of scoliosis using a finite element model". *European Spine Journal,* **19**, pp. 1169–1178.

[34] Trease, B. P., Moon, Y., and Kota, S., 2005. "Design of large-displacement compliant joints". *Journal of Mechanical Design,* **127**(4), pp. 788–798.

[35] Ball, R. S., 1876. *The Theory of Screws: A Study in the Dynamics of a Rigid Body.* Hodges, Foster, and Co., Dublin.

[36] Body Intelligence Training, 2013. Spine landmarks, September.

[37] Schmid, S., Studer, D., Hasler, C.-C., Romkes, J., Taylor, W. R., Brunner, R., and Lorenzetti, S., 2015. "Using skin markers for spinal curvature quantification in main thoracic adolescent idiopathic scoliosis: An explorative radiographic study". *PloS one,* **10**(8).

[38] Lipkin, H., and Patterson, T., 1992. "Geometric properties of modelled robot elasticity: Part i - decomposition". In Robotics, spatial mechanisms, and mechanical systems, Vol. 45, American Society of Mechanical Engineers.

[39] Lipkin, H., and Patterson, T., 1992. "Geometric properties of modelled robot elasticity: Part ii - center of elasticity". In Robotics, spatial mechanisms, and mechanical systems, Vol. 45, American Society of Mechanical Engineers.

[40] Cibiak, N., and Lipkin, H., 1994. "Centers of stiffness, compliance, and elasticity in the modelling of robotics systems". In Robotics: Kinematics, Dynamics and Controls, Vol. 72, American Society of Mechanical Engineers.

[41] Hopkins, J. B., 2005. "Design of parallel flexure systems via freedom and constraint topologies (fact)". PhD thesis, Massachusetts Institute of Technology.

[42] Davidson, J. K., and Hunt, K. H., 2004. *Robots and Screw Theory, Applications of kinematics and statics to robotics,.* Oxford University Press.

[43] Nijssen, J., 2016. "A type synthesis approach to compliant shell mechanisms". Master's thesis, Delft University of Technology.

Appendix A

General Method for Deriving Compliance Ellipsoid

The compliance matrix needed to be determined for mechanisms so that the characteristic ellipsoids could be determined. In the available Finite Element (FE) software, COMSOL Multiphysics, the compliance and stiffness matrices were not easily accessible. To derive the compliance matrix, the mechanism was displaced by a unit force and moment in each direction, one at a time. Since the compliance ellipsoid is a linear representation, the unit force and moment magnitudes were required to be just enough to displace the mechanism a small amount. The 6 degrees of freedom of displacement were recorded for each load case. From each load case, one column of the compliance matrix could be calculated. Once the full compliance matrix was obtained, the eigenvalue decomposition from the previous section was performed.

Consider applying a unit force in the x-direction, $F = \begin{bmatrix} F_x & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ resulting in the displacement $U = \begin{bmatrix} \Delta x & \Delta y & \Delta z & \Delta\theta_x & \Delta\theta_y & \Delta\theta_z \end{bmatrix}^T$. Then, as can be

extrapolated from Equation A.1, $C_{11} = \frac{\Delta x}{F_x}$, $C_{21} = \frac{\Delta y}{F_x}$, and so on. Every column of the compliance matrix can be filled in this manner, providing the result needed to find the compliance ellipsoids of a mechanism.

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\theta_x \\ \Delta\theta_y \\ \Delta\theta_z \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{bmatrix} F_x \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.1}$$

Since the ellipsoids provide a linear description for a mechanism only at the rest point, a nonlinear solver was not necessary for this analysis. In addition, the applied unit force was only required to be large enough to cause a small deflection. In this case, the Euler-Bernoulli beam model was used because only small deformations were expected and because the beams were long and slender.

Appendix B

Determing Ideal Beam Thickness

B.1 Finite Element Interface

In the designing the brace, MATLAB, COMSOL, and SolidWorks were used in combination through the LiveLink interface. An STL file of the successful brace design provided by collaborators was imported into SolidWorks where the brace shell was modified and the beams of mechanisms were defined. The mechanism beam dimensions and locations were controlled using parameters that were changed in COMSOL via the LiveLink connection. Also using LiveLink, the beam geometry was imported into COMSOL. Once in COMSOL, the general FE settings were changed, and forces and constraints were applied. After the setup, performing the COMSOL analyses were controlled by MATLAB through LiveLink. MATLAB was capable of changing beam dimensions in SolidWorks, updating the beam geometry in COMSOL, performing the analyses, and storing results. The results were then utilized for useful purposes such as determining the compliance ellipsoid for a mechanism or finding ideal beam

thicknesses.

B.2 Ideal Beam Thickness Criterion

Table B.1 describes the process of finding the ideal beam thickness for the cross-pivot mechanisms using phosphor bronze. At 0.9mm beam thickness, the maximum angle difference before reaching fatigue strength of 400 MPa is 1.4° with only 1N of force. At 1.75 mm of thickness, 50N of force does not cause the mechanism to reach the fatigue strength. Somewhere between these is the ideal thickness, where the most motion is achieved. It was determined that the mechanisms can achieve the most rotational motion when the beams are thickest without reaching their fatigue strength when the 50N load is applied. It was found that 1.58 mm thickness was ideal, but it only allowed 11.8° of rotation. This is far below the requirement of 30.8°. For this reason, other materials were sought.

Table B.1: Phosphor bronze manual optimization example.

Beam Width (mm)	**Force (N)**	**Angle (°)**	**Stress (MPa)**	**Angle Diff(°)**
0.9	0	21.5	377	
	1	22.9	397.25	1.4
1	0	15.1	305	
	5	20.2	373	5.1
1.2	0	8.7	195	
	20	19.7	420	11
1.5	0	4.6	129	
	40	15.6	377	11
1.75	0	3	98	
	50	11.7	321	8.7
1.58	0	4	118	
	50	15.8	398	11.8

Appendix C

Previous Work

A previous iteration of design was presented in IDETC2016-59616, A Passive Brace to Improve Activities of Daily Living Utilizing Compliant Parallel Mechanisms. This previous iteration utilized the screw analysis without clustering and averaging. The force requirements used were purely estimates from literature. The brace that was designed was a proof-of-concept that compliant mechanisms were capable of the necessary stiffness.

Figure 3.9 shows the final kinematic requirement from this paper, a set of screws. The outliers were eliminated by only plotting a small set of the screws that were clustered. From this, it was known that the motion occurred about a lateral axis. Since the screws did not remain in one position, it was assumed they the mechanism needed translation in addition to rotation. Therefore, the mechanism in Figure C.1 was developed and manufactured using Delrin.

Similarly to the current design, the old design was tested for stiffness by displacing

Figure C.1: Mechanism designed for flexion in previous iteration.

it and measuring the forces applied. It was determined that this mechanism was capable of 5.4 N/mm stiffness, which was greater the stiffness, 1.2 N/mm, that had been found through literature. Therefore, this design proof-of-concept has been deemed plausible with respect to kinetic requirements.

The kinematic qualification was determined by having the subject wear the brace and perform the motion, as seen in Figure C.2. It can be seen that the mechanism deflected when the subject bends.

Figure C.2: Subject performing flexion, the motion for which the mechanism was designed.

This previous design was a good proof-of-concept, but more detail was added in this thesis to develop a brace that better corrects the spine and follows the subject's motion.

Appendix D

MATLAB Code

The code used to perform the various analyses of this thesis are presented in the sections below. Each analysis contains one primary code, but may also include auxiliary codes.

D.1 Screw Analysis

The *screwAnalysis_main* file contains the main code used to perform the screw analysis. The screw parameters were calculated in *findTransformation*. All other codes performed auxiliary functions needed to perform the analysis.

```
clear all
close all
clc

%% setup
fileLocations = '../Vicon Data/20160627Michelle';


fileName = 'MaxBend.csv';

rangeMotion = [1,2152];
frameFreq = 100;
nRows = 2152;
time = linspace(0,nRows/frameFreq,nRows);

baseAcol = 41;
baseBcol = baseAcol-3;
baseCcol = baseAcol+3;

set1Acol = 32;
set1Bcol = set1Acol-3;
set1Ccol = set1Acol+3;

set2Acol = 23;
set2Bcol = set2Acol-3;
set2Ccol = set2Acol+3;

set3Acol = 14;
set3Bcol = set3Acol-3;
set3Ccol = set3Acol+3;

set4Acol = 5;
set4Bcol = set4Acol-3;
set4Ccol = set4Acol+3;

%% Intialize and read data
% AnalysisSetup
cd(fileLocations)

% READ DATA FROM FILE

%base
baseB = csvread(fileName,5,baseBcol,[5,baseBcol,nRows+4,baseBcol+2]);
baseC = csvread(fileName,5,baseCcol,[5,baseCcol,nRows+4,baseCcol+2]);
baseA = csvread(fileName,5,baseAcol,[5,baseAcol,nRows+4,baseAcol+2]);

% sets about base where 1 is first about base and 4 is the top set
set1B = csvread(fileName,5,set1Bcol,[5,set1Bcol,nRows+4,set1Bcol+2]);
set1C = csvread(fileName,5,set1Ccol,[5,set1Ccol,nRows+4,set1Ccol+2]);
set1A = csvread(fileName,5,set1Acol,[5,set1Acol,nRows+4,set1Acol+2]);
```

```
set2B = csvread(fileName,5,set2Bcol,[5,set2Bcol,nRows+4,set2Bcol+2]);
set2C = csvread(fileName,5,set2Ccol,[5,set2Ccol,nRows+4,set2Ccol+2]);
set2A = csvread(fileName,5,set2Acol,[5,set2Acol,nRows+4,set2Acol+2]);

set3B = csvread(fileName,5,set3Bcol,[5,set3Bcol,nRows+4,set3Bcol+2]);
set3C = csvread(fileName,5,set3Ccol,[5,set3Ccol,nRows+4,set3Ccol+2]);
set3A= csvread(fileName,5,set3Acol,[5,set3Acol,nRows+4,set3Acol+2]);

set4B = csvread(fileName,5,set4Bcol,[5,set4Bcol,nRows+4,set4Bcol+2]);
set4C = csvread(fileName,5,set4Ccol,[5,set4Ccol,nRows+4,set4Ccol+2]);
set4A= csvread(fileName,5,set4Acol,[5,set4Acol,nRows+4,set4Acol+2]);

% cd '..'
cd '..'
cd '../Kinematic Analysis'

%% switch data
tempA = baseA;
tempB = baseB;
tempC = baseC;

baseA = set4A;
baseB = set4B;
baseC = set4C;

set4A = tempA;
set4B = tempB;
set4C = tempC;

tempA = set1A;
tempB = set1B;
tempC = set1C;

set1A = set3A;
set1B = set3B;
set1C = set3C;

set3A = tempA;
set3B = tempB;
set3C = tempC;

%% Clean data
badRows = [];
nbr=0;
for i = 1:size(baseB)
    if baseA(i,1)+baseA(i,2)+baseA(i,3) == 0 || baseB(i,1)+baseB(i,2)+baseB(i,3) == 0 ||↙
baseC(i,1)+baseC(i,2)+baseC(i,3) == 0 ||...
            set1A(i,1)+set1A(i,2)+set1A(i,3) == 0 || set1B(i,1)+set1B(i,2)+set1B(i,3) ==↙
0 || set1C(i,1)+set1C(i,2)+set1C(i,3) == 0 ||...
            set2A(i,1)+set2A(i,2)+set2A(i,3) == 0 || set2B(i,1)+set2B(i,2)+set2B(i,3) ==↙
0 || set2C(i,1)+set2C(i,2)+set2C(i,3) == 0 ||...
```

```
            set3A(i,1)+set3A(i,2)+set3A(i,3) == 0 || set3B(i,1)+set3B(i,2)+set3B(i,3) ==↙
0 || set3C(i,1)+set3C(i,2)+set3C(i,3) == 0 ||...
            set4A(i,1)+set4A(i,2)+set4A(i,3) == 0 || set4B(i,1)+set4B(i,2)+set4B(i,3) ==↙
0 || set4C(i,1)+set4C(i,2)+set4C(i,3) == 0
        nbr = nbr+1;
        badRows = [i,badRows];
    end
end

histogram(badRows,nRows)

[~,numBadRows] = size(badRows);
badRowsTxt = ['There are ', num2str(numBadRows), ' out of ', num2str(nRows), '. Continue?↙
Y/N?\n'];
fprintf(badRowsTxt);
% response = input(badRowsTxt,'s');
% if response == 'N'
%     disp('Program Terminated')
%     break  %#ok<BRKCONT>
% end

%
baseB(badRows,:) = []; baseC(badRows,:) = []; baseA(badRows,:) = [];
set1B(badRows,:) = []; set1C(badRows,:) = []; set1A(badRows,:) = [];
set2B(badRows,:) = []; set2C(badRows,:) = []; set2A(badRows,:) = [];
set3B(badRows,:) = []; set3C(badRows,:) = []; set3A(badRows,:) = [];
set4B(badRows,:) = []; set4C(badRows,:) = []; set4A(badRows,:) = [];

rangeMotion(1) = 1;
[rangeMotion(2),~]= size(baseB);
nRows = rangeMotion(2);

%% Convert sets to rigid bodies

instance0 = rangeMotion(1);

[baseA,baseB,baseC] = makeRigidBody(instance0,baseA,baseB,baseC);
[set1A,set1B,set1C] = makeRigidBody(instance0,set1A,set1B,set1C);
[set2A,set2B,set2C] = makeRigidBody(instance0,set2A,set2B,set2C);
[set3A,set3B,set3C] = makeRigidBody(instance0,set3A,set3B,set3C);
[set4A,set4B,set4C] = makeRigidBody(instance0,set4A,set4B,set4C);

%% Rotate so that base(1) is in xy-plane

[baseA,baseB,baseC,set1A,set1B,set1C,set2A,set2B,set2C,set3A,set3B,set3C,set4A,set4B,↙
set4C] = ...
    transformToXYplane( [0,0,0],[1,0,0],[0,1,0], baseB(instance0,1:3), baseC(instance0,1:↙
3), baseA(instance0,1:3), ...
    baseA,baseB,baseC,set1A,set1B,set1C,set2A,set2B,set2C,set3A,set3B,set3C,set4A,set4B,↙
set4C);
```

```
%% Rotate so that all bases are in same position
frameSpacing = 1;


for i = 1:nRows

    [baseBinBase(i,:),baseCinBase(i,:),baseAinBase(i,:),set1BinBase(i,:),set1CinBase↙
(i,:),set1AinBase(i,:), ...
        set2BinBase(i,:),set2CinBase(i,:),set2AinBase(i,:),set3BinBase(i,:),set3CinBase↙
(i,:),set3AinBase(i,:),...
        set4BinBase(i,:),set4CinBase(i,:),set4AinBase(i,:)] = ...
        transformToPlane4( 0,baseB(instance0,:), baseC(instance0,:),  baseA(instance0,:),↙
baseB(i,:), baseC(i,:), baseA(i,:), ...
        baseB(i,:),baseC(i,:),baseA(i,:),set1B(i,:),set1C(i,:),set1A(i,:),set2B(i,:),↙
set2C(i,:),set2A(i,:),...
        set3B(i,:),set3C(i,:),set3A(i,:),set4B(i,:),set4C(i,:),set4A(i,:));
end
beep


figure(16)
hold on; axis image
begin = 500;
fin = 600;
x = [baseAinBase(begin:fin,1),set1AinBase(begin:fin,1),set2AinBase(begin:fin,1),↙
set3AinBase(begin:fin,1),set4AinBase(begin:fin,1)];
y = [baseAinBase(begin:fin,2),set1AinBase(begin:fin,2),set2AinBase(begin:fin,2),↙
set3AinBase(begin:fin,2),set4AinBase(begin:fin,2)];
z = [baseAinBase(begin:fin,3),set1AinBase(begin:fin,3),set2AinBase(begin:fin,3),↙
set3AinBase(begin:fin,3),set4AinBase(begin:fin,3)];

plot3(x',y',z')

%  set 1 to set 2

for i = 1:nRows
    [set1Ain1(i,:),set1Bin1(i,:),set1Cin1(i,:), set2Ain1(i,:),set2Bin1(i,:),set2Cin1↙
(i,:),...
        set3Ain1(i,:),set3Bin1(i,:),set3Cin1(i,:), set4Ain1(i,:),set4Bin1(i,:),set4Cin1↙
(i,:)] = ...
        transformToPlane4( 0, ...
        set1BinBase(instance0,:), set1CinBase(instance0,:), set1AinBase(instance0,:),↙
set1BinBase(i,:), set1CinBase(i,:), set1AinBase(i,:), ...
        set1AinBase(i,:),set1BinBase(i,:),set1CinBase(i,:),set2AinBase(i,:),set2BinBase↙
(i,:),set2CinBase(i,:),...
        set3AinBase(i,:),set3BinBase(i,:),set3CinBase(i,:),set4AinBase(i,:),set4BinBase↙
(i,:),set4CinBase(i,:));
end
beep

%  set 2 to set 3
```

```
for i = 1:nRows
    [set2Ain2(i,:),set2Bin2(i,:),set2Cin2(i,:),...
        set3Ain2(i,:),set3Bin2(i,:),set3Cin2(i,:), set4Ain2(i,:),set4Bin2(i,:),set4Cin2↙
(i,:)] = ...
        transformToPlane4( 0, ...
        set2Ain1(instance0,:),set2Bin1(instance0,:), set2Cin1(instance0,:), set2Ain1↙
(i,:), set2Bin1(i,:), set2Cin1(i,:), ...
        set2Ain1(i,:),set2Bin1(i,:),set2Cin1(i,:), set3Ain1(i,:),set3Bin1(i,:),set3Cin1↙
(i,:),set4Ain1(i,:),set4Bin1(i,:),set4Cin1(i,:));
end

%  set 3 to set 4

for i = 1:nRows
    [set3Ain3(i,:),set3Bin3(i,:),set3Cin3(i,:), set4Ain3(i,:),set4Bin3(i,:),set4Cin3↙
(i,:)] = ...
        transformToPlane4( 0, ...
        set3Ain2(instance0,:),set3Bin2(instance0,:), set3Cin2(instance0,:), set3Ain2↙
(i,:), set3Bin2(i,:), set3Cin2(i,:), ...
        set3Ain2(i,:),set3Bin2(i,:),set3Cin2(i,:),set4Ain2(i,:),set4Bin2(i,:),set4Cin2↙
(i,:));
end

%% move to easy visual
[baseAinBase,baseBinBase,baseCinBase,set1AinBase,set1BinBase,set1CinBase,set1Ain1,↙
set1Bin1,set1Cin1, ...
        set2Ain1,set2Bin1,set2Cin1,set2Ain2,set2Bin2,set2Cin2,set3Ain2,set3Bin2,set3Cin2,↙
set3Ain3,set3Bin3,set3Cin3,...
        set4Ain3,set4Bin3,set4Cin3] = ...
        transformToPlane2([0,-baseAinBase(instance0,2),0],[1,-baseAinBase(instance0,2),↙
0],[0,-baseAinBase(instance0,2)+1,0],baseBinBase(instance0,:),baseAinBase(instance0,:),↙
baseCinBase(instance0,:),...
        baseAinBase,baseBinBase,baseCinBase,set1AinBase,set1BinBase,set1CinBase,set1Ain1,↙
set1Bin1,set1Cin1, ...
        set2Ain1,set2Bin1,set2Cin1,set2Ain2,set2Bin2,set2Cin2,set3Ain2,set3Bin2,set3Cin2,↙
set3Ain3,set3Bin3,set3Cin3,...
        set4Ain3,set4Bin3,set4Cin3);

%% Find Screws
frameSpacing = 1;

backLen10 = norm(set1C(instance0,:)-baseC(instance0,:));
backLen20 = norm(set2C(instance0,:)-set1C(instance0,:));
backLen30 = norm(set3C(instance0,:)-set2C(instance0,:));
backLen40 = norm(set4C(instance0,:)-set3C(instance0,:));

for i = rangeMotion(1)+frameSpacing:frameSpacing:rangeMotion(2)

    % calc changes in distance of A
    backLen1c = norm(set1C(i,:)-baseC(i,:));
```

```
    backLen1(i) =  backLen1c - backLen10;

    backLen2c = norm(set2C(i,:)-set1C(i,:));
    backLen2(i) =  backLen2c - backLen20;

    backLen3c = norm(set3C(i,:)-set2C(i,:));
    backLen3(i) =  backLen3c - backLen30;

    backLen4c = norm(set4C(i,:)-set3C(i,:));
    backLen4(i) =  backLen4c - backLen40;

    %    screws
    [Tset1(:,:,i),thetaset1(i),wset1(i,:),qset1(i,:),hset1(i)] = ...
        findTransformation(set1AinBase(i,:),set1BinBase(i,:),set1CinBase(i,:),set1AinBase↙
(instance0+1,:),set1BinBase(instance0+1,:),set1CinBase(instance0+1,:));

    [Tset2(:,:,i),thetaset2(i),wset2(i,:),qset2(i,:),hset2(i)] = ...
        findTransformation(set2Ain1(i,:),set2Bin1(i,:),set2Cin1(i,:),set2Ain1↙
(instance0+1,:),set2Bin1(instance0+1,:),set2Cin1(instance0+1,:));

    [Tset3(:,:,i),thetaset3(i),wset3(i,:),qset3(i,:),hset3(i)] = ...
        findTransformation(set3Ain2(i,:),set3Bin2(i,:),set3Cin2(i,:),set3Ain2↙
(instance0+1,:),set3Bin2(instance0+1,:),set3Cin2(instance0+1,:));

    [Tset4(:,:,i),thetaset4(i),wset4(i,:),qset4(i,:),hset4(i)] = ...
        findTransformation(set4Ain3(i,:),set4Bin3(i,:),set4Cin3(i,:),set4Ain3↙
(instance0+1,:),set4Bin3(instance0+1,:),set4Cin3(instance0+1,:));

    [Tset2Base(:,:,i),thetaset2Base(i),wset2Base(i,:),qset2Base(i,:),hset2Base(i)] = ...
        findTransformation(set2AinBase(i,:),set2BinBase(i,:),set2CinBase(i,:),set2AinBase↙
(instance0+1,:),set2BinBase(instance0+1,:),set2CinBase(instance0+1,:));

end

figure(15)
% suptitle('Sagittal Bending')

subplot(4,1,1)
plot(time,backLen1)
title('Set 1')
xlabel('Time (seconds)')
ylabel('Extension (mm)')

subplot(4,1,2)
plot(time,backLen2)
title('Set 2')
xlabel('Time (seconds)')
ylabel('Extension (mm)')

subplot(4,1,3)
plot(time,backLen3)
```

```
title('Set 3')
xlabel('Time (seconds)')
ylabel('Extension (mm)')

subplot(4,1,4)
plot(time,backLen4)
title('Set 4 ')
xlabel('Time (seconds)')
ylabel('Extension (mm)')

%% Modify screw starting points
normLine1 = [0,1,0]; %-cross(set1BinBase(instance0,:)-set1AinBase(instance0,:),↙
set1CinBase(instance0,:)-set1AinBase(instance0,:));
for i = 1:nRows
   [qset1mod(i,1:3),check(i)] = plane_line_intersect(normLine1,set1AinBase(instance0,:),↙
qset1(i,:),qset1(i,:)+10000*wset1(i,:));
end

normLine2 = [0,1,0]; %-cross(set2Bin1(instance0,:)-set2Ain1(instance0,:),set2Cin1↙
(instance0,:)-set2Ain1(instance0,:));
for i = 1:nRows
   [qset2mod(i,1:3),check(i)] = plane_line_intersect(normLine2,set2Ain1(instance0,:),↙
qset2(i,:),qset2(i,:)+10000*wset2(i,:));
end

normLine3 = [0,1,0]; %-cross(set3Bin2(instance0,:)-set3Ain2(instance0,:),set3Cin2↙
(instance0,:)-set3Ain2(instance0,:));
for i = 1:nRows
   [qset3mod(i,1:3),check(i)] = plane_line_intersect(normLine3,set3Ain2(instance0,:),↙
qset3(i,:),qset3(i,:)+10000*wset3(i,:));
end

normLine4 = [0,1,0]; %-cross(set4Bin3(instance0,:)-set4Ain3(instance0,:),set4Cin3↙
(instance0,:)-set4Ain3(instance0,:));
for i = 1:nRows
   [qset4mod(i,1:3),check(i)] = plane_line_intersect(normLine4,set4Ain3(instance0,:),↙
qset4(i,:),qset4(i,:)+10000*wset4(i,:));
end

%% SAVE DATA TO FILE
cd(fileLocations)

save('MaxBend')

%% Plot all sets and screws.

TWIST = 0;
FULL = 1;

figure(1111); clf;hold on; axis image
view(3)
```

```
% view(0,90)
title('Lower Set Frame')
xlabel('X (mm)');ylabel('Y (mm)');zlabel('Z (mm)');

XbaseChosen = [baseAinBase(instance0,1),baseBinBase(instance0,1),baseCinBase(instance0,↙
1)];
YbaseChosen = [baseAinBase(instance0,2),baseBinBase(instance0,2),baseCinBase(instance0,↙
2)];
ZbaseChosen = [baseAinBase(instance0,3),baseBinBase(instance0,3),baseCinBase(instance0,↙
3)];
patch(XbaseChosen,YbaseChosen,ZbaseChosen,[0 0 0]);


frameSpacing = 5;
l=1;

if TWIST == 1
    for i = rangeMotion(1)+frameSpacing:frameSpacing:rangeMotion(2)

        Xbase(:,i) = [baseA(i,1),baseB(i,1),baseC(i,1)];% ,DpelvisIN4(i,1),C1in4(i,1)];
        Ybase(:,i) = [baseA(i,2),baseB(i,2),baseC(i,2)];%,DpelvisIN4(i,2),C1in4(i,2)];
        Zbase(:,i) = [baseA(i,3),baseB(i,3),baseC(i,3)];%,DpelvisIN4(i,3),C1in4(i,3)];
        patch(Xbase(:,i),Ybase(:,i),Zset1(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);

        Xset1(:,i) = [set1B(i,1),set1C(i,1)];%set1A(i,1),set1B(i,1),set1C(i,1)];% ,↙
DpelvisIN4(i,1),C1in4(i,1)];
        Yset1(:,i) = [set1B(i,2),set1C(i,2)];%set1A(i,2),set1B(i,2),set1C(i,2)];%,↙
DpelvisIN4(i,2),C1in4(i,2)];
        Zset1(:,i) = [set1B(i,3),set1C(i,3)];%set1A(i,3),set1B(i,3),set1C(i,3)];%,↙
DpelvisIN4(i,3),C1in4(i,3)];
        patch(Xset1(:,i),Yset1(:,i),Zset1(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);

        Xset2(:,i) = [set2B(i,1),set2C(i,1)];%set2A(i,1),set2B(i,1),set2C(i,1)];% ,↙
DpelvisIN4(i,1),C1in4(i,1)];
        Yset2(:,i) = [set2B(i,2),set2C(i,2)];%set2A(i,2),set2B(i,2),set2C(i,2)];%,↙
DpelvisIN4(i,2),C1in4(i,2)];
        Zset2(:,i) = [set2B(i,3),set2C(i,3)];%set2A(i,3),set2B(i,3),set2C(i,3)];%,↙
DpelvisIN4(i,3),C1in4(i,3)];
        patch(Xset2(:,i),Yset2(:,i),Zset2(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);

        Xset3(:,i) = [set3B(i,1),set3C(i,1)];%set3A(i,1),set3B(i,1),set3C(i,1)];% ,↙
DpelvisIN4(i,1),C1in4(i,1)];
        Yset3(:,i) = [set3B(i,2),set3C(i,2)];%set3A(i,2),set3B(i,2),set3C(i,2)];%,↙
DpelvisIN4(i,2),C1in4(i,2)];
        Zset3(:,i) = [set3B(i,3),set3C(i,3)];%set3A(i,3),set3B(i,3),set3C(i,3)];%,↙
DpelvisIN4(i,3),C1in4(i,3)];
        patch(Xset3(:,i),Yset3(:,i),Zset3(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);
```

```
        Xset4(:,i) = [set4B(i,1),set4C(i,1)];%set4A(i,1),set4B(i,1),set4C(i,1)];% ,↙
DpelvisIN4(i,1),C1in4(i,1)];
        Yset4(:,i) = [set4B(i,2),set4C(i,2)];%set4A(i,2),set4B(i,2),set4C(i,2)];%,↙
DpelvisIN4(i,2),C1in4(i,2)];
        Zset4(:,i) = [set4B(i,3),set4C(i,3)];%set4A(i,3),set4B(i,3),set4C(i,3)];%,↙
DpelvisIN4(i,3),C1in4(i,3)];
        patch(Xset4(:,i),Yset4(:,i),Zset4(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);
    end
end

if FULL == 1
    for i = rangeMotion(1)+frameSpacing:frameSpacing:rangeMotion(2)

        Xbase(:,i) = [baseAinBase(i,1),baseBinBase(i,1),baseCinBase(i,1)];%baseA(i,1),↙
baseB(i,1),baseC(i,1)];% ,DpelvisIN4(i,1),C1in4(i,1)];
        Ybase(:,i) = [baseAinBase(i,2),baseBinBase(i,2),baseCinBase(i,2)];%baseA(i,2),↙
baseB(i,2),baseC(i,2)];%,DpelvisIN4(i,2),C1in4(i,2)];
        Zbase(:,i) = [baseAinBase(i,3),baseBinBase(i,3),baseCinBase(i,3)];%baseA(i,3),↙
baseB(i,3),baseC(i,3)];%,DpelvisIN4(i,3),C1in4(i,3)];
        patch(Xbase(:,i),Ybase(:,i),Zbase(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);

%                 Xset1(:,i) = [set1AinBase(i,1),set1BinBase(i,1),set1CinBase(i,1)];% ,↙
DpelvisIN4(i,1),C1in4(i,1)];
%                 Yset1(:,i) = [set1AinBase(i,2),set1BinBase(i,2),set1CinBase(i,2)];%,↙
DpelvisIN4(i,2),C1in4(i,2)];
%                 Zset1(:,i) = [set1AinBase(i,3),set1BinBase(i,3),set1CinBase(i,3)];%,↙
DpelvisIN4(i,3),C1in4(i,3)];
%                 patch(Xset1(:,i),Yset1(:,i),Zset1(:,i),[0 (i-rangeMotion(1))/↙
(rangeMotion(2)-rangeMotion(1)) 0]);
%
        Xset1(:,i) = [set1Ain1(i,1),set1Bin1(i,1),set1Cin1(i,1)];% ,DpelvisIN4(i,1),C1in4↙
(i,1)];
        Yset1(:,i) = [set1Ain1(i,2),set1Bin1(i,2),set1Cin1(i,2)];%,DpelvisIN4(i,2),C1in4↙
(i,2)];
        Zset1(:,i) = [set1Ain1(i,3),set1Bin1(i,3),set1Cin1(i,3)];%,DpelvisIN4(i,3),C1in4↙
(i,3)];
        patch(Xset1(:,i),Yset1(:,i),Zset1(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);

%                 Xset2(:,i) = [set2Ain1(i,1),set2Bin1(i,1),set2Cin1(i,1)];% ,DpelvisIN4↙
(i,1),C1in4(i,1)];
%                 Yset2(:,i) = [set2Ain1(i,2),set2Bin1(i,2),set2Cin1(i,2)];%,DpelvisIN4↙
(i,2),C1in4(i,2)];
%                 Zset2(:,i) = [set2Ain1(i,3),set2Bin1(i,3),set2Cin1(i,3)];%,DpelvisIN4↙
(i,3),C1in4(i,3)];
%                 patch(Xset2(:,i),Yset2(:,i),Zset2(:,i),[0 (i-rangeMotion(1))/↙
(rangeMotion(2)-rangeMotion(1)) 0]);
```

```
        Xset2(:,i) = [set2Ain2(i,1),set2Bin2(i,1),set2Cin2(i,1)];% ,DpelvisIN4(i,1),C1in4
(i,1)];
        Yset2(:,i) = [set2Ain2(i,2),set2Bin2(i,2),set2Cin2(i,2)];%,DpelvisIN4(i,2),C1in4
(i,2)];
        Zset2(:,i) = [set2Ain2(i,3),set2Bin2(i,3),set2Cin2(i,3)];%,DpelvisIN4(i,3),C1in4
(i,3)];
        patch(Xset2(:,i),Yset2(:,i),Zset2(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-
rangeMotion(1)) 0]);
        %
%                       Xset3(:,i) = [set3Ain2(i,1),set3Bin2(i,1),set3Cin2(i,1)];% ,
DpelvisIN4(i,1),C1in4(i,1)];
%                       Yset3(:,i) = [set3Ain2(i,2),set3Bin2(i,2),set3Cin2(i,2)];%,
DpelvisIN4(i,2),C1in4(i,2)];
%                       Zset3(:,i) = [set3Ain2(i,3),set3Bin2(i,3),set3Cin2(i,3)];%,
DpelvisIN4(i,3),C1in4(i,3)];
%                       patch(Xset3(:,i),Yset3(:,i),Zset3(:,i),[0 (i-rangeMotion(1))/
(rangeMotion(2)-rangeMotion(1)) 0]);

        Xset3(:,i) = [set3Ain3(i,1),set3Bin3(i,1),set3Cin3(i,1)];% ,DpelvisIN4(i,1),C1in4
(i,1)];
        Yset3(:,i) = [set3Ain3(i,2),set3Bin3(i,2),set3Cin3(i,2)];%,DpelvisIN4(i,2),C1in4
(i,2)];
        Zset3(:,i) = [set3Ain3(i,3),set3Bin3(i,3),set3Cin3(i,3)];%,DpelvisIN4(i,3),C1in4
(i,3)];
        patch(Xset3(:,i),Yset3(:,i),Zset3(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-
rangeMotion(1)) 0]);

        %
        Xset4(:,i) = [set4Ain3(i,1),set4Bin3(i,1),set4Cin3(i,1)];% ,DpelvisIN4(i,1),C1in4
(i,1)];
        Yset4(:,i) = [set4Ain3(i,2),set4Bin3(i,2),set4Cin3(i,2)];%,DpelvisIN4(i,2),C1in4
(i,2)];
        Zset4(:,i) = [set4Ain3(i,3),set4Bin3(i,3),set4Cin3(i,3)];%,DpelvisIN4(i,3),C1in4
(i,3)];
        patch(Xset4(:,i),Yset4(:,i),Zset4(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-
rangeMotion(1)) 0]);

        E = 500;
        mean1 = mean(qset1mod,'omitnan');
        if abs(mean1(1)-qset1mod(i-frameSpacing,1)) < E && abs(mean1(2)-qset1mod(i-
frameSpacing,2)) < E && abs(mean1(3)-qset1mod(i-frameSpacing,3)) < E
            quiver3(qset1mod(i-frameSpacing,1),qset1mod(i-frameSpacing,2),qset1mod(i-
frameSpacing,3),wset1(i-frameSpacing,1),wset1(i-frameSpacing,2),wset1(i-frameSpacing,3),
75,'Color',[(i-rangeMotion(1))/(rangeMotion(2)-rangeMotion(1)) 0 0])%*thetaset1(i-
frameSpacing));
        end
        %         quiver3(qset1(i-frameSpacing,1),qset1(i-frameSpacing,2),qset1(i-
frameSpacing,3),wset1(i-frameSpacing,1),wset1(i-frameSpacing,2),wset1(i-frameSpacing,3),
100,'Color',[(i-rangeMotion(1))/(rangeMotion(2)-rangeMotion(1)) 0 0])%*thetaset1(i-
frameSpacing));
```

```
        mean2 = mean(qset2mod,'omitnan');
        if abs(mean2(1)-qset2mod(i-frameSpacing,1)) < E && abs(mean2(2)-qset2mod(i-frameSpacing,2)) < E && abs(mean2(3)-qset2mod(i-frameSpacing,3)) < E
            quiver3(qset2mod(i-frameSpacing,1),qset2mod(i-frameSpacing,2),qset2mod(i-frameSpacing,3),wset2(i-frameSpacing,1),wset2(i-frameSpacing,2),wset2(i-frameSpacing,3),75,'Color',[0 (i-rangeMotion(1))/(rangeMotion(2)-rangeMotion(1)) 0])%*thetaset2(i-frameSpacing));
        end

        mean3 = mean(qset3mod,'omitnan');
        if abs(mean3(1)-qset3mod(i-frameSpacing,1)) < E && abs(mean3(2)-qset3mod(i-frameSpacing,2)) < E && abs(mean3(3)-qset3mod(i-frameSpacing,3)) < E
            quiver3(qset3mod(i-frameSpacing,1),qset3mod(i-frameSpacing,2),qset3mod(i-frameSpacing,3),wset3(i-frameSpacing,1),wset3(i-frameSpacing,2),wset3(i-frameSpacing,3),75,'Color',[0 0 (i-rangeMotion(1))/(rangeMotion(2)-rangeMotion(1))])%*thetaset3(i-frameSpacing));
        end

        mean4 = mean(qset4mod,'omitnan');
        if abs(mean4(1)-qset4mod(i-frameSpacing,1)) < E && abs(mean4(2)-qset4mod(i-frameSpacing,2)) < E && abs(mean4(3)-qset4mod(i-frameSpacing,3)) < E
            quiver3(qset4mod(i-frameSpacing,1),qset4mod(i-frameSpacing,2),qset4mod(i-frameSpacing,3),wset4(i-frameSpacing,1),wset4(i-frameSpacing,2),wset4(i-frameSpacing,3),75,'Color',[(i-rangeMotion(1))/(rangeMotion(2)-rangeMotion(1)) 0 0])%*thetaset4(i-frameSpacing));
        end
        %
        %M(l) = getframe;
        l=l+1;
        %     if i < rangeMotion(2)-frameSpacing
        %         delete(p)
        %     end
    end
    %movie(M);
    %movie2avi(M,'mid with respect to lower')

end

%% Plot Screw Rotation


DISP_ROT = 1;

if DISP_ROT == 1
    fprintf('Max Rotation about Screw\nset1: %f \nset2: %f \nset3: %f \nset4: %f \nSet2Base for Validation: %f \n', ...
        max(abs(thetaset1))*180/pi,max(abs(thetaset2))*180/pi,max(abs(thetaset3))*180/pi,max(abs(thetaset4))*180/pi,max(abs(thetaset2Base))*180/pi)

    fprintf('Difference Rotation about Screw\nset1: %f \nset2: %f \nset3: %f \nset4: %f \nSet2Base for Validation: %f \n', ...
```

```
        (abs(max(thetaset1))+abs(min(thetaset1)))*180/pi, ...
        (abs(max(thetaset2))+abs(min(thetaset2)))*180/pi, ...
        (abs(max(thetaset3))+abs(min(thetaset3)))*180/pi, ...
        (abs(max(thetaset4))+abs(min(thetaset4)))*180/pi, ...
        (abs(max(thetaset2Base))+abs(min(thetaset2Base)))*180/pi)

end


figure(12)
hold on
title('Twist')
hold off

subplot(4,1,1)
plot(time,abs(thetaset1)*180/pi)
title('Set 1')
xlabel('Time (seconds)')
ylabel('Rotation Angle of Motion (degrees)')

subplot(4,1,2)
plot(time,abs(thetaset2)*180/pi)
title('Set 2')
xlabel('Time (seconds)')
ylabel('Rotation Angle of Motion (degrees)')

subplot(4,1,3)
plot(time,abs(thetaset3)*180/pi)
title('Set 3')
xlabel('Time (seconds)')
ylabel('Rotation Angle of Motion (degrees)')

subplot(4,1,4)
plot(time,abs(thetaset4*180)/pi)
title('Set 4 ')
xlabel('Time (seconds)')
ylabel('Rotation Angle of Motion (degrees)')

%%
figure(20); clf;hold on; axis image
frameSpacing = 1;
view(3)
% view(0,90)
title('Lower Set Frame')
xlabel('X (mm)');ylabel('Y (mm)');zlabel('Z (mm)');

for i = rangeMotion(1)+frameSpacing:frameSpacing:rangeMotion(2)

    Xbase(:,i) = [baseAinBase(i,1),baseBinBase(i,1),baseCinBase(i,1)];% ,DpelvisIN4(i,1),↙
C1in4(i,1)];
    Ybase(:,i) = [baseAinBase(i,2),baseBinBase(i,2),baseCinBase(i,2)];%,DpelvisIN4(i,2),↙
```

```
C1in4(i,2)];
    Zbase(:,i) = [baseAinBase(i,3),baseBinBase(i,3),baseCinBase(i,3)];%,DpelvisIN4(i,3),↙
C1in4(i,3)];
    patch(Xbase(:,i),Ybase(:,i),Zbase(:,i),[0 (i-rangeMotion(1))/(rangeMotion(2)-↙
rangeMotion(1)) 0]);
end
```

```
% [T,w,q,h] = findTransformation(A,B,C)
% Calculates the homogenous transformation, T, based on the points A,B,C
% using A0 as the point of reference.
%
% The function also returns the screw parameters
%       w (free vector in direction of rotation axis),
%       q (point on rotation axis),
%       h (pitch of screw)
%    A,B,C are all [nx3]. First row is assumed to be the reference from
%    which all transformations are found.

function [T,theta,w,q,h] = findTransformation(A,B,C,A0,B0,C0)

u = B0 - A0;  u = u/norm(u);
T(:,:,1) = eye(4);

for count = 1:size(B,1)

        Btilde = (B(count,:) - A(count,:))/norm(B(count,:)-A(count,:)) - ...
        (B0-A0)/norm(B0-A0);
    Ctilde = (C(count,:) - A(count,:))/norm(C(count,:)-A(count,:)) - ...
         (C0-A0)/norm(C0-A0);

    w(count,:) = cross(Ctilde,Btilde);
    w(count,:) = w(count,:)/norm(w(count,:));
    v = B(count,:) - A(count,:); v = v/norm(v);
    uprime = u - w(count,:)*dot(w(count,:),u);
    vprime = v - w(count,:)*dot(w(count,:),v);


    theta(count) = atan2(dot(w(count,:),cross(uprime,vprime)),...
        dot(uprime,vprime));

    R = rodrigues(w(count,:),theta(count));
    d =  A(count,:)' - R*A0';  %A(count,:)' - A0';
    T(:,:,count) = [R, d;zeros(1,3),1];

    h(count) = dot(w(count,:),d')/theta(count);
    dNormal = d' - h(count)*theta(count)*w(count,:);
    %isItNormal = dot(dNormal,w(count,:))
    q(count,:) = (dNormal/2 + cross(w(count,:),dNormal)/2/tan(theta(count)/2)); %%
end


% returns the rotation matrix formed by the axis w and angle theta based on
% the Rodrigues formulation.
function R = rodrigues(w,theta)
wHat = [0,-w(3),w(2); w(3),0,-w(1); -w(2),w(1),0];
R = eye(3) + wHat*sin(theta) + wHat^2*(1-cos(theta));
```

```
function [ J, Knew, Lnew ] = makeRigidBody( instance0, J, K, L )
%makeRigidBody converts the every instance of JKL to be the same distance
%relative to each other. It uses instance0 to determine the proper
%distances to use.
%   J,K,L - a column of x,y,z locations, respectively
%   J, Knew, Lnew - new JKL with adjusted distances.

JKmagnitude = norm(K(instance0,:)-J(instance0,:));
JLmagnitude = norm(L(instance0,:)-J(instance0,:));
KJLangle = acos((norm(K(instance0,:)-J(instance0,:))^2+norm(L(instance0,:)-J↙
(instance0,:))^2-norm(L(instance0,:)-K(instance0,:))^2)/(2*norm(K(instance0,:)-J↙
(instance0,:))*norm(L(instance0,:)-J(instance0,:))));


for i = 1:size(J,1)
    JKunit = (K(i,:)-J(i,:))/norm(K(i,:)-J(i,:));
    JLunit = (L(i,:)-J(i,:))/norm(L(i,:)-J(i,:));

    KJLanglei = acos((norm(K(i,:)-J(i,:))^2+norm(L(i,:)-J(i,:))^2-norm(L(i,:)-K(i,:))^2)/↙
(2*norm(K(i,:)-J(i,:))*norm(L(i,:)-J(i,:))));

    axisRot = -cross(JKunit,JLunit); axisRot = axisRot/norm(axisRot);

    JKunitnew = JLunit*cos(KJLangle) + cross(axisRot,JLunit) + axisRot*dot(axisRot,↙
JLunit)*(1-cos(KJLangle));
    JKunitnew = JKunitnew/norm(JKunitnew);

    Knew(i,:) = J(i,:) + (JKunitnew*JKmagnitude);
    Lnew(i,:) = J(i,:) + (JLunit*JLmagnitude);

end

end
```

```
function [I,check]=plane_line_intersect(n,V0,P0,P1)
%plane_line_intersect computes the intersection of a plane and a segment(or
%a straight line)
% Inputs:
%       n: normal vector of the Plane
%       V0: any point that belongs to the Plane
%       P0: end point 1 of the segment P0P1
%       P1:  end point 2 of the segment P0P1
%
%Outputs:
%      I    is the point of interection
%     Check is an indicator:
%      0 => disjoint (no intersection)
%      1 => the plane intersects P0P1 in the unique point I
%      2 => the segment lies in the plane
%      3=>the intersection lies outside the segment P0P1
%
% Example:
% Determine the intersection of following the plane x+y+z+3=0 with the segment P0P1:
% The plane is represented by the normal vector n=[1 1 1]
% and an arbitrary point that lies on the plane, ex: V0=[1 1 -5]
% The segment is represented by the following two points
% P0=[-5 1 -1]
%P1=[1 2 3]
% [I,check]=plane_line_intersect([1 1 1],[1 1 -5],[-5 1 -1],[1 2 3]);

%This function is written by :
%                             Nassim Khaled
%                             Wayne State University
%                             Research Assistant and Phd candidate
%If you have any comments or face any problems, please feel free to leave
%your comments and i will try to reply to you as fast as possible.

I=[0 0 0];
u = P1-P0;
w = P0 - V0;
D = dot(n,u);
N = -dot(n,w);
check=0;
if abs(D) < 10^-7        % The segment is parallel to plane
        if N == 0           % The segment lies in plane
            check=2;
            return
        else
            check=0;       %no intersection
            return
        end
end

%compute the intersection parameter
sI = N / D;
```

```
I = P0+ sI.*u;

if (sI < 0 || sI > 1)
    check= 3;          %The intersection point  lies outside the segment, so there is no↙
intersection
else
    check=1;
end
```

```
function [ varargout ] = transformToPlane2( pop1, pop2, pop3, p1, p2, p3, varargin)
%   Inputs
    % pop1, pop2, pop3 (points on plane) - [x,y,z] point used to define plane to rotate
    % data into
    % p1,p2,p3 - points to reference for rotation
    % varargin (data to rotate) - data points to be transformed. varargin
    % allows a variable number of input, allowing for any number of points
    % to be rotated by this function

d = p1 - pop1;

p1A = p1-d;
p2A = p2-d;
p3A = p3-d;

r1 = vrrotvec(p2A-p1A,pop2-pop1);
k1 = r1(1:3);
theta1 = r1(4);

p2B = p1A + (p2A-p1A)*cos(theta1) + cross(k1,(p2A-p1A))*sin(theta1) + k1*dot(k1,(p2A-↙
p1A))*(1-cos(theta1));
p3B = p1A + (p3A-p1A)*cos(theta1) + cross(k1,(p3A-p1A))*sin(theta1) + k1*dot(k1,(p3A-↙
p1A))*(1-cos(theta1));

r2 = vrrotvec(p3B-p1A,pop3-pop1);

k2 = r2(1:3);
theta2 = r2(4);

p2C = p1A + (p2B-p1A)*cos(theta2) + cross(k2,(p2B-p1A))*sin(theta2) + k2*dot(k2,(p2B-↙
p1A))*(1-cos(theta2));
p3C = p1A + (p3B-p1A)*cos(theta2) + cross(k2,(p3B-p1A))*sin(theta2) + k2*dot(k2,(p3B-↙
p1A))*(1-cos(theta2));

for kArgs = 1:length(varargin)
    DTRin = varargin{kArgs};
    for kDTR = 1:size(DTRin)
        DTR = DTRin(kDTR,:);

        DTR1(kDTR,:) = DTR-d;
        DTR2(kDTR,:) = p1A + (DTR1(kDTR,:) - p1A)*cos(theta1) + cross(k1,(DTR1(kDTR,:) -↙
p1A))*sin(theta1) + k1*dot(k1,(DTR1(kDTR,:) - p1A))*(1-cos(theta1));
        DTR3(kDTR,:) = p1A + (DTR2(kDTR,:) - p1A)*cos(theta2) + cross(k2,(DTR2(kDTR,:) -↙
p1A))*sin(theta2) + k2*dot(k2,(DTR2(kDTR,:) - p1A))*(1-cos(theta2));
        DTRout(kDTR,:) = DTR3(kDTR,:);
    end
    varargout{kArgs} = DTRout;
end
end
```

```
function [ varargout ] = transformToPlane4( thetaPlus, pop1, pop2, pop3, p1, p2, p3,↙
varargin)
%   Inputs
    % pop1, pop2, pop3 (points on plane) - [x,y,z] point used to define plane to rotate
    % data into
    % p1,p2,p3 - points to reference for rotation
    % varargin (data to rotate) - data points to be transformed. varargin
    % allows a variable number of input, allowing for any number of points
    % to be rotated by this function

d = p1 - pop1;

p1A = p1-d;
p2A = p2-d;
p3A = p3-d;

r1 = vrrotvec(p2A-p1A,pop2-pop1);
k1 = r1(1:3);
theta1 = r1(4);

p2B = p1A + (p2A-p1A)*cos(theta1) + cross(k1,(p2A-p1A))*sin(theta1) + k1*dot(k1,(p2A-↙
p1A))*(1-cos(theta1));
p3B = p1A + (p3A-p1A)*cos(theta1) + cross(k1,(p3A-p1A))*sin(theta1) + k1*dot(k1,(p3A-↙
p1A))*(1-cos(theta1));

closest = findClosestLineToPoint( p2B, p1A, p3B);

r2 = vrrotvec(p3B-closest,pop3-closest);
k2 = r2(1:3);
theta2 = r2(4)-thetaPlus;

for kArgs = 1:length(varargin)
    DTRin = varargin{kArgs};

    for kDTR = 1:size(DTRin)

        DTR = DTRin(kDTR,:);
        DTR1(kDTR,:) = DTR-d;
        DTR2(kDTR,:) = p1A + (DTR1(kDTR,:) - p1A)*cos(theta1) + cross(k1,(DTR1(kDTR,:) -↙
p1A))*sin(theta1) + k1*dot(k1,(DTR1(kDTR,:) - p1A))*(1-cos(theta1));
        DTR3(kDTR,:) = p1A + (DTR2(kDTR,:) - p1A)*cos(theta2) + cross(k2,(DTR2(kDTR,:) -↙
p1A))*sin(theta2) + k2*dot(k2,(DTR2(kDTR,:) - p1A))*(1-cos(theta2));
        DTRout(kDTR,:) = DTR3(kDTR,:);
    end
    varargout{kArgs} = DTRout;
end

end
```

```
function [ varargout ] = transformToXYplane( pop1, pop2, pop3, p1, p2, p3, varargin)
%   Inputs
    % pop1, pop2, pop3 (points on plane) - [x,y,z] point used to define plane to rotate
    % data into
    % p1,p2,p3 - points to reference for rotation
    % varargin (data to rotate) - data points to be transformed. varargin
    % allows a variable number of input, allowing for any number of points
    % to be rotated by this function

d = p1 - pop1;

p1A = p1-d;
p2A = p2-d;
p3A = p3-d;

r1 = vrrotvec(p2A-p1A,pop2-pop1);
k1 = r1(1:3);
theta1 = r1(4);

p2B = p1A + (p2A-p1A)*cos(theta1) + cross(k1,(p2A-p1A))*sin(theta1) + k1*dot(k1,(p2A-↙
p1A))*(1-cos(theta1));
p3B = p1A + (p3A-p1A)*cos(theta1) + cross(k1,(p3A-p1A))*sin(theta1) + k1*dot(k1,(p3A-↙
p1A))*(1-cos(theta1));

p3Bmod = [0,p3B(1,2),p3B(1,3)];
p1Amod = [0,p1A(1,2),p1A(1,3)];
pop3mod = [0,pop3(1,2),pop3(1,3)];
pop1mod = [0,pop1(1,2),pop1(1,3)];

r2 = vrrotvec(p3Bmod-p1Amod,pop3mod-pop1mod);
k2 = r2(1:3);
theta2 = r2(4)+pi;

p2C = p1A + (p2B-p1A)*cos(theta2) + cross(k2,(p2B-p1A))*sin(theta2) + k2*dot(k2,(p2B-↙
p1A))*(1-cos(theta2));
p3C = p1A + (p3B-p1A)*cos(theta2) + cross(k2,(p3B-p1A))*sin(theta2) + k2*dot(k2,(p3B-↙
p1A))*(1-cos(theta2));

for kArgs = 1:length(varargin)
    DTRin = varargin{kArgs};
    for kDTR = 1:size(DTRin)

        DTR = DTRin(kDTR,:);
        DTR1(kDTR,:) = DTR-d;
        DTR2(kDTR,:) = p1A + (DTR1(kDTR,:) - p1A)*cos(theta1) + cross(k1,(DTR1(kDTR,:) -↙
p1A))*sin(theta1) + k1*dot(k1,(DTR1(kDTR,:) - p1A))*(1-cos(theta1));
        DTR3(kDTR,:) = p1A + (DTR2(kDTR,:) - p1A)*cos(theta2) + cross(k2,(DTR2(kDTR,:) -↙
p1A))*sin(theta2) + k2*dot(k2,(DTR2(kDTR,:) - p1A))*(1-cos(theta2));
        DTRout(kDTR,:) = DTR3(kDTR,:);

    end
```

```
    varargout{kArgs} = DTRout;
end

end
```

D.2 Ellipsoid Generation with COMSOL Interface

The code below was used to produce the characterizations for the library of mechanisms presented in this thesis.

```
%% Compliance Elliptoid calculator
% by JB Ring and Joep Nijssen
% Program uses input of a COMSOL created finite element model to determine
% the compliance matrix. Extensions include the calculation of the
% eigenwrench and eigentwists.

function complianceMatrix = Compliance_Elliptioid_calculator_func(modelName)
% allow models to be imported from the COMSOL interface
import com.comsol.model.*
import com.comsol.model.util.*

% Restart program
% close all
% clear all
% clc

% input the COMSOL model into the program
model = mphload(modelName)

%% get parameter names
parameter_names = model.param.varnames()

%% set all force/moment for first set
% add unit load/ moment in all direction on actuation point, the point is
% then evaluated on displacements and rotations.
model.param.set('ForceX', '1[N]');
model.param.set('ForceY', '0[N]');
model.param.set('ForceZ', '0[N]');
model.param.set('MomentX', '0[N*mm]');
model.param.set('MomentY', '0[N*mm]');
model.param.set('MomentZ', '0[N*mm]');

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Fx = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('ForceX', '0[N]');
model.param.set('ForceY', '1[N]');

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Fy = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];
```

```
%
model.param.set('ForceY', '0[N]');
model.param.set('ForceZ', '1[N]');

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Fz = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('ForceZ', '0[N]');
model.param.set('MomentX', '1000[N*mm]');

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Mx = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('MomentX', '0[N*mm]');
model.param.set('MomentY', '1000[N*mm]');

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_My = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('MomentY', '0[N*mm]');
model.param.set('MomentZ', '1000[N*mm]');

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Mz = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%% find Compliance Matrix
unitForce = 1; %Newton
```

```
unitMoment = 1; %Newton-mm

DispRot_Fx

C(:,1) = DispRot_Fx./unitForce;
C(:,2) = DispRot_Fy./unitForce;
C(:,3) = DispRot_Fz./unitForce;
C(:,4) = DispRot_Mx./unitMoment;
C(:,5) = DispRot_My./unitMoment;
C(:,6) = DispRot_Mz./unitMoment;
% C is the complete compliance matrix


% divide compliance matrix up into its ABD decomposition
A=C(1:3, 1:3);
B1=C(1:3,4:6); % off diagonal matrices
B2=C(4:6,1:3)*1000; % off diagonal matrices
D=C(4:6,4:6);

% format entries of matrix into short notation
format short g
% return matrix back to full complaince matrix
C=[A B1; B2 D];


% because of very small numerical errors we filter the matrix. Values which
% are seen as small compared to the main compliance values are set to 0.
% Also the small numerical differences between entries is set equal by
% summation and division by 2.
C = round(C,4);
for m = 1:6
    for n = m+1:6
        if C(m,n) ~= C(n,m)
            ave = mean([C(m,n),C(n,m)]);
            C(m,n) = ave;
            C(n,m) = ave;
        end
    end
end

%% Linear Compliance elliptoid Calculator

% Decomposition back into it's ABD form.
A=C(1:3, 1:3);
B1=C(1:3,4:6); % off diagonal matrices
B2=C(4:6,1:3); % off diagonal matrices
D=C(4:6,4:6);

%% determine eigenvectors and eigenvalues
[vCf,eigCf]=eig(A);
[vCm,eigCm]=eig(D);
```

```
% eigenvectors linear translation compliance matrix
vC1=vCf(:,1);
vC2=vCf(:,2);
vC3=vCf(:,3);

% eigenvectors linear rotation compliance matrix
vC1m=vCm(:,1);
vC2m=vCm(:,2);
vC3m=vCm(:,3);

%% checks on matrices
[~,p] = chol(A) % check if matrix is positive-definite. If p=0 then true.
% check on invertibility of matrix
det(A) %if Det= nonzero then invertible

% % determine angle between eigenvectors
% ThetaInDegrees = atan2d(norm(cross(vlcs3,vlcs1)),dot(vlcs3,vlcs1))

%% unit sphere of force
% creating the unit sphere of loading
syms lambda zeta


xsp=0.5*cos(zeta)*sin(lambda)';
ysp=0.5*sin(zeta)*sin(lambda)';
zsp=0.5*cos(lambda)';

V=[xsp; ysp; zsp];

% multiplications of stiffness matrix with unit sphere force
Cellips=A*V;
CellipsG=D*V;

%vector multiplier
a=0.01;

% vector multiplier geometric nonline
a1=0.0001;

ellipsoidFigure = figure(1);
title('Different corrugated shell elliptoids')
hold on
subplot(1,2,1);
ezsurf(Cellips(1),Cellips(2),Cellips(3),[-pi,pi,-pi,pi],40)
alpha(0.0001)
hold on
vec1=quiver3(0,0,0,vC1(1),vC1(2),vC1(3),a);
set(vec1,'linewidth',4);
hold on
vec2=quiver3(0,0,0,vC2(1),vC2(2),vC2(3),a);
```

```
set(vec2,'linewidth',4);
hold on
vec3=quiver3(0,0,0,vC3(1),vC3(2),vC3(3),a);
set(vec3,'linewidth',4);
hold on
axis image
title('Compliance ellips linear translation matrix')
subplot(1,2,2);
ezsurf(CellipsG(1),CellipsG(2),CellipsG(3),[-pi,pi,-pi,pi],40)
alpha(0.0001)
hold on
vec11=quiver3(0,0,0,vC1m(1),vC1m(2),vC1m(3),a1);
set(vec11,'linewidth',4);
hold on
vec22=quiver3(0,0,0,vC2m(1),vC2m(2),vC2m(3),a1);
set(vec22,'linewidth',4);
hold on
vec33=quiver3(0,0,0,vC3m(1),vC3m(2),vC3m(3),a1);
set(vec33,'linewidth',4);
axis image
title('Compliance ellips rotational matrix')

%% Save Figure and Data
savefig(ellipsoidFigure,strcat(modelName(1:length(modelName)-4),'.fig'))
clear model
save(modelName(1:length(modelName)-4))
```

D.3 Eigentwist Generation with COMSOL Interface

The *getEigenTwist* code determines the eigentwist of a mechanism using a input COMSOL model. The other two codes are auxiliary functions needed to manipulate the COMSOL model.

```
function [ eigenScrew ] = getEigenTwist(model,force, toPlot)
%getEigentwist determines the eigentwist for a mechanism in the import
%model by applying the unit force.
%   model - input comsol model
%   force - unit force magnitude
%   toPlot - turns on/off plot
%   eigenScrew - output 6x1 matrix for screw parameters of the eigentwist

%% setup
import com.comsol.model.*
import com.comsol.model.util.*

unitForce = 1; %Newton
unitMoment = 1; %Newton-m

%% get parameter names
% parameter_names = model.param.varnames()

%% add unit load/ moment in all direction on actuation point, the point is
% set all force/moment for first set
model.param.set('ForceX', strcat(num2str(force),'[N]'));
model.param.set('ForceY', '0[N]');
model.param.set('ForceZ', '0[N]');
model.param.set('MomentX', '0[N*mm]');
model.param.set('MomentY', '0[N*mm]');
model.param.set('MomentZ', '0[N*mm]');

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Fx = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('ForceX', '0[N]');
model.param.set('ForceY', strcat(num2str(force),'[N]'));

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Fy = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('ForceY', '0[N]');
model.param.set('ForceZ', strcat(num2str(force),'[N]'));
```

```
model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Fz = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('ForceZ', '0[N]');
model.param.set('MomentX', strcat(num2str(force),'[N*mm]'));

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Mx = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('MomentX', '0[N*mm]');
model.param.set('MomentY', strcat(num2str(force),'[N*mm]'));

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_My = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%
model.param.set('MomentY', '0[N*mm]');
model.param.set('MomentZ', strcat(num2str(force),'[N*mm]'));

model.sol('sol1').runAll;
model.result('pg1').run;

DispRot_Mz = [model.result.numerical('pev1').getReal(),model.result.numerical('pev2').↙
getReal(),model.result.numerical('pev3').getReal(), ...
    model.result.numerical('pev4').getReal(), model.result.numerical('pev5').getReal(),↙
model.result.numerical('pev6').getReal()];

%% Determine compliance matrix

% DispRot_Fx

C(:,1) = DispRot_Fx./unitForce;
C(:,2) = DispRot_Fy./unitForce;
C(:,3) = DispRot_Fz./unitForce;
```

```
C(:,4) = DispRot_Mx./unitMoment;
C(:,5) = DispRot_My./unitMoment;
C(:,6) = DispRot_Mz./unitMoment;
Cor = C;

% because of very small numerical errors we filter the matrix to garuntee
% symmetry. Values which are seen as small compared to the main compliance
% values are set to 0. Also the small numerical differences between entries
% is set equal by summation and division by 2.
C = round(C,10);
for m = 1:6
    for n = m+1:6
        if C(m,n) ~= C(n,m)
            ave = mean([C(m,n),C(n,m)]);
            C(m,n) = ave;
            C(n,m) = ave;
        end
    end
end

%% Determine EigenTwist
Ac=C(1:3, 1:3); %K11
B1c=C(1:3,4:6); %K12
B2c=C(4:6,1:3); %K21
Dc=C(4:6,4:6); %K22

[yil,lambdacl]=eigs(Dc);       %lambdacl=eigenvaluesmatrix
lambda1c=lambdacl(1,1);
lambda2c=lambdacl(2,2);
lambda3c=lambdacl(3,3);

% eigenvectors
y1=yil(:,1);
y2=yil(:,2);
y3=yil(:,3);


% sigma values - translation
sigma1=(1/lambda1c)*B1c*y1; % (1/eigenvalue)*B*translation
sigma2=(1/lambda2c)*B1c*y2;
sigma3=(1/lambda3c)*B1c*y3;

% eigentwists are (sigmai = translation; yi = rotation)
eigentwist1=[sigma1;y1];
eigentwist2=[sigma2;y2];
eigentwist3=[sigma3;y3];

eigentwist1=round(eigentwist1,7);
eigentwist2=round(eigentwist2,7);
eigentwist3=round(eigentwist3,7);
```

```
eigentwistsmatrix=[eigentwist1, eigentwist2, eigentwist3];

% determine the 3 rotational vector of the eigentwists and 3 translational
% components of eigentwist
eigentwistrotation1=[eigentwist1(4:6)];
eigentwistrotation2=[eigentwist2(4:6)];
eigentwistrotation3=[eigentwist3(4:6)];

eigentwisttranslation1=[eigentwist1(1:3)];
eigentwisttranslation2=[eigentwist2(1:3)];
eigentwisttranslation3=[eigentwist3(1:3)];

%% for location
% determine orthogonality between rotational components
ThetaInDegrees1 = atan2d(norm(cross(eigentwistrotation1,eigentwistrotation2)),dot↙
(eigentwistrotation1,eigentwistrotation2));
ThetaInDegrees2 = atan2d(norm(cross(eigentwistrotation1,eigentwistrotation3)),dot↙
(eigentwistrotation1,eigentwistrotation3));
ThetaInDegrees3 = atan2d(norm(cross(eigentwistrotation2,eigentwistrotation3)),dot↙
(eigentwistrotation2,eigentwistrotation3));

% pitch value
% determining the pitch value h of eigentwist
h1=dot(eigentwistrotation1,eigentwisttranslation1)/dot(eigentwistrotation1,↙
eigentwistrotation1);
h2=dot(eigentwistrotation2,eigentwisttranslation2)/dot(eigentwistrotation2,↙
eigentwistrotation2);
h3=dot(eigentwistrotation3,eigentwisttranslation3)/dot(eigentwistrotation3,↙
eigentwistrotation3);


% solving the linear system for each eigentwist

% locations with respect to [0,0,0]
X1 = cross(eigentwistrotation1,(eigentwisttranslation1-h1*eigentwistrotation1))/dot↙
(eigentwistrotation1,eigentwistrotation1);

X2 = cross(eigentwistrotation2,(eigentwisttranslation2-h2*eigentwistrotation2))/dot↙
(eigentwistrotation2,eigentwistrotation2);

X3 = cross(eigentwistrotation3,(eigentwisttranslation3-h3*eigentwistrotation3))/dot↙
(eigentwistrotation3,eigentwistrotation3);

rotation1 = [eigentwist1(4,:) eigentwist1(5,:) eigentwist1(6,:)]'; %abs([eigentwist1(4,:)↙
eigentwist1(5,:) eigentwist1(6,:)])';
rotation2 = [eigentwist2(4,:) eigentwist2(5,:) eigentwist2(6,:)]'; %abs([eigentwist2(4,:)↙
eigentwist2(5,:) eigentwist2(6,:)])';
rotation3 = [eigentwist3(4,:) eigentwist3(5,:) eigentwist3(6,:)]'; %abs([eigentwist3(4,:)↙
eigentwist3(5,:) eigentwist3(6,:)])';

pitch1=dot(eigentwistrotation1,eigentwisttranslation1)/dot(eigentwistrotation1,↙
```

```
eigentwistrotation1);
pitch2=dot(eigentwistrotation2,eigentwisttranslation2)/dot(eigentwistrotation2,↙
eigentwistrotation2);
pitch3=dot(eigentwistrotation3,eigentwisttranslation3)/dot(eigentwistrotation3,↙
eigentwistrotation3);

sig1=[sigma1(1), sigma1(2), sigma1(3)];

if toPlot
    figure
    hold on
    quiver3(X1(1),X1(2),X1(3),rotation1(1),rotation1(2),rotation1(3))
    quiver3(X2(1),X2(2),X2(3),rotation2(1),rotation2(2),rotation2(3))
    quiver3(X3(1),X3(2),X3(3),rotation3(1),rotation3(2),rotation3(3))
    xlabel('X')
    ylabel('Y')
    hold off
end

eigenScrew = [X1(1),X1(2),X1(3),rotation1(1),rotation1(2),rotation1(3);...
    X2(1),X2(2),X2(3),rotation2(1),rotation2(2),rotation2(3);...
    X3(1),X3(2),X3(3),rotation3(1),rotation3(2),rotation3(3)];

end
```

```
function [ complete, CSLL, CSLR ] = resetBeamOrientation3(model,leftVertex,rightVertex)
%resetBeamOrientation changes the current beam  width direction such that
%it is perpendicular to the plane of the mechanism.
%   model - imported comsol model
%   leftVertex - the vertex that makes a line perpendicular to the left
%   mechanism
%   rightVertex - the vertex that makes a line perpendicular to the right
%   mechanism
%   complete - returns that function was completed
%   CSLL - the location of the left alignment point
%   CSLR - the location of the right alignment point

model.geom('geom1').measureFinal.selection.geom('geom1', 0);
model.geom('geom1').measureFinal.selection.set(leftVertex);
CSLL = model.geom('geom1').measureFinal.getVtxCoord();

% model.geom('geom1').measureFinal.selection.geom('geom1', 0);
model.geom('geom1').measureFinal.selection.set(rightVertex);
CSLR = model.geom('geom1').measureFinal.getVtxCoord();

model.param.set('CS_L_x', [num2str(CSLL(1)),'[mm]']);
model.param.set('CS_L_y', [num2str(CSLL(2)),'[mm]']);
model.param.set('CS_L_z', [num2str(CSLL(3)),'[mm]']);
model.param.set('CS_R_x', [num2str(CSLR(1)),'[mm]']);
model.param.set('CS_R_y', [num2str(CSLR(2)),'[mm]']);
model.param.set('CS_R_z', [num2str(CSLR(3)),'[mm]']);
% CSLL(3)

complete = 1;
end
```

```
function [ complete ] = varyMech_mm( model, position, side, yBwidthT, yFwidthB, zB, zT,↙
beamThick, planeThick,mechAngle)
%varyMech_mm is used to vary the dimensions of the mechanisms through the
%comsol/matlab/solidworks interface, LiveLink.
% model - imported comsol model
% position - where on the body the mechanism is located
% side - side of the body on which the mechanism is located
% other inputs - parameters used to control mechanism dimensions



%   if any value is 0, do not change it.
switch position

    case 'low'
        if side == 'L'
            if yBwidthT ~= 0;
                model.param.set('LL_CC_LL_yB_Sketch16', [num2str(yBwidthT),'[mm]']);
            end;
            if yFwidthB ~= 0; model.param.set('LL_CC_LL_yF_Sketch16', [num2str↙
(yFwidthB),'[mm]']); end;
            if zB ~= 0; model.param.set('LL_CC_LL_zB_Sketch16', [num2str(zB),'[mm]']);↙
end;
            if zT ~= 0; model.param.set('LL_CC_LL_zT_Sketch16', [num2str(zT),'[mm]']);↙
end;
            if beamThick ~= 0; model.param.set('CC_LL_beamThick', [num2str↙
(beamThick),'[mm]']); end;
            if planeThick ~= 0; model.param.set('CC_LL_planarThick', [num2str↙
(planeThick),'[mm]']); end;
            if mechAngle >= 0; model.param.set('LL_CC_LL_mechAngle', [num2str↙
(mechAngle),'[deg]']); end;
        elseif side == 'R'
            if yBwidthT ~= 0; model.param.set('LL_CC_LR_yB_Sketch17', [num2str↙
(yBwidthT),'[mm]']); end;
            if yFwidthB ~= 0; model.param.set('LL_CC_LR_yF_Sketch17', [num2str↙
(yFwidthB),'[mm]']); end;
            if zB ~= 0; model.param.set('LL_CC_LR_zB_Sketch17', [num2str(zB),'[mm]']);↙
end;
            if zT ~= 0; model.param.set('LL_CC_LR_zT_Sketch17', [num2str(zT),'[mm]']);↙
end;
            if beamThick ~= 0; model.param.set('CC_LR_beamThick', [num2str↙
(beamThick),'[mm]']); end;
            if planeThick ~= 0; model.param.set('CC_LR_planarThick', [num2str↙
(planeThick),'[mm]']); end;
            if mechAngle >= 0; model.param.set('LL_CC_LR_mechAngle', [num2str↙
(mechAngle),'[deg]']); end;
        else
            error('invalid side choice')
        end
        %%
    case 'low2'
```

```
        if side == 'L'
            if yBwidthT ~= 0;
                model.param.set('LL_CC_LL_yB_Sketch16', [num2str(yBwidthT),'[mm]']);
            end;
            if yFwidthB ~= 0; model.param.set('LL_CC_LL_yF_Sketch16', [num2str↙
(yFwidthB),'[mm]']); end;
            if zB ~= 0; model.param.set('LL_CC_LL_zB_Sketch16', [num2str(zB),'[mm]']);↙
end;
            if zT ~= 0; model.param.set('LL_CC_LL_zT_Sketch16', [num2str(zT),'[mm]']);↙
end;
            if beamThick ~= 0; model.param.set('CC_L_beamThick', [num2str↙
(beamThick),'[mm]']); end;
            if planeThick ~= 0; model.param.set('CC_L_planarThick', [num2str↙
(planeThick),'[mm]']); end;
            if mechAngle >= 0; model.param.set('LL_CC_LL_mechAngle', [num2str↙
(mechAngle),'[deg]']); end;
        elseif side == 'R'
            if yBwidthT ~= 0; model.param.set('LL_CC_LR_yB_Sketch17', [num2str↙
(yBwidthT),'[mm]']); end;
            if yFwidthB ~= 0; model.param.set('LL_CC_LR_yF_Sketch17', [num2str↙
(yFwidthB),'[mm]']); end;
            if zB ~= 0; model.param.set('LL_CC_LR_zB_Sketch17', [num2str(zB),'[mm]']);↙
end;
            if zT ~= 0; model.param.set('LL_CC_LR_zT_Sketch17', [num2str(zT),'[mm]']);↙
end;
            if beamThick ~= 0; model.param.set('CC_L_beamThick', [num2str↙
(beamThick),'[mm]']); end;
            if planeThick ~= 0; model.param.set('CC_L_planarThick', [num2str↙
(planeThick),'[mm]']); end;
            if mechAngle >= 0; model.param.set('LL_CC_LR_mechAngle', [num2str↙
(mechAngle),'[deg]']); end;
        else
            error('invalid side choice')
        end
        %%
    case 'mid'
        if side == 'F'
            if yBwidthT ~= 0;model.param.set('CC_MF_widthT', [num2str(yBwidthT),'[mm]']);↙
end;
            if yFwidthB ~= 0; model.param.set('CC_MF_widthB', [num2str↙
(yFwidthB),'[mm]']); end;
            if zB ~= 0; model.param.set('CC_MF_zB', [num2str(zB),'[mm]']); end;
            if zT ~= 0; model.param.set('CC_MF_zT', [num2str(zT),'[mm]']); end;
            if beamThick ~= 0; model.param.set('CC_LL_beamThick', [num2str↙
(beamThick),'[mm]']); end;
            if planeThick ~= 0; model.param.set('CC_LL_planarThick', [num2str↙
(planeThick),'[mm]']); end;
            %             if mechAngle >= 0; model.param.set('LL_CC_LL_mechAngle',↙
[num2str(mechAngle),'[deg]']); end;
        elseif side == 'B'
            if yBwidthT ~= 0;model.param.set('CC_MB_widthT', [num2str(yBwidthT),'[mm]']);↙
```

```
end;
            if yFwidthB ~= 0; model.param.set('CC_MB_widthB', [num2str↙
(yFwidthB),'[mm]']); end;
            if zB ~= 0; model.param.set('CC_MB_zB', [num2str(zB),'[mm]']); end;
            if zT ~= 0; model.param.set('CC_MB_zT', [num2str(zT),'[mm]']); end;
            if beamThick ~= 0; model.param.set('CC_LR_beamThick', [num2str↙
(beamThick),'[mm]']); end;
            if planeThick ~= 0; model.param.set('CC_LR_planarThick', [num2str↙
(planeThick),'[mm]']); end;
            %            if mechAngle >= 0; model.param.set('LL_CC_LR_mechAngle',↙
[num2str(mechAngle),'[deg]']); end;
        else
            error('invalid side choice')
        end

end
complete = 1;
end
```

D.4 Interface for Force/Displacement Curve

The forces from the physical force/displacement test were input into the COMSOL model using *ForceDisplacementPlotComsolInterface* to find the theoretical force-displacement curves for each mechanism.

```
import com.comsol.model.*
import com.comsol.model.util.*

addpath('C:\Users\jbr024\Google Drive\1.↙
Scoliosis\Cases\Montreal_withPressure\FEA\relatedFunctions')
addpath('C:\Users\jbr024\Google Drive\1. Scoliosis\Cases\Montreal_withPressure\FEA')

% Restart program
close all
clearvars -except model comsolpwd % clear all
clc

%% input the COMSOL model into the program

model = mphload('Brace_Design4_CrossPivot_forViconCompare');

studyName = 'ForceY_Good';
fileName = studyName; %regexprep(studyName,' ','');
savename = 'forceY_good2_data.mat';


%% adjust mechanisms
%varyMech( model, position, side,
%                          yB, yF, zB, zT, beamThick, planeThick,angle)
varyMech_mm(model,'low2','L', 200, 127.9762, 30, 45, 1.2, 4,8);
varyMech_mm(model,'low2','R', 200, 127.9762, 30, 45, 1.2, 4,8);

%                           widthT, widthB, zB, zT, beamThick, planeThick,mechAngle)
varyMech_mm( model,'mid','F', 32,   160,    43, 60, .78, 12,0);
varyMech_mm( model,'mid','B', 62,   140,    28, 60, .78, 12,0);

model.geom('geom1').feature('cad1').updateCadParamTable(false, false);
model.geom('geom1').feature('cad1').importData;
model.geom('geom1').run;
%
%
%%%% FIX THESE %%%%
% [ complete, CSLL, CSLR ] = resetBeamOrientation3(model,29,5);   % cartwheel 13,1);
% %leftFront,rightFront, leftBack, rightBack)
% resetBeamOrientation2(model,19,14, 23,12)
%%%%%%%%%%%%%%%%%%%%

%% study with Vicon force input

%%%
load('translation_y_2_lower_good.mat'); range = [1586,1779];

model.param.set('ForceX', '0[N]'); %strcat(num2str(force),'[N]'));
model.param.set('ForceY', '0[N]');
model.param.set('ForceZ', '0[N]');
model.param.set('MomentX', '0[N*m]');
```

```
model.param.set('MomentY', '0[N*m]');
model.param.set('MomentZ', '0[N*m]');

tic
count = 0;
forRange = range(1):round((range(2)-range(1))/15,0):range(2);
for i = forRange(2:end)
    count = count + 1

    model.param.set('ForceX', strcat(num2str(-forceAdj(i,1)),'[N]'));
    model.param.set('ForceY', strcat(num2str(-forceAdj(i,2)),'[N]'));
    model.param.set('ForceZ', strcat(num2str(-forceAdj(i,3)),'[N]'));

    model.param.set('MomentX', strcat(num2str(-momentAdj(i,1)),'[N*m]'));
    model.param.set('MomentY', strcat(num2str(-momentAdj(i,2)),'[N*m]'));
    model.param.set('MomentZ', strcat(num2str(-momentAdj(i,3)),'[N*m]'));
%
    model.sol('sol1').runAll;
    model.result('pg1').run;
%
    DispRot_F1(count,:) = [model.result.numerical('pev1').getReal(),model.result.↙
numerical('pev2').getReal(),model.result.numerical('pev3').getReal(), ...
        model.result.numerical('pev4').getReal(), model.result.numerical('pev5').↙
getReal(),model.result.numerical('pev6').getReal()]
    Indexes1(count) = i;
end

%%%%

save(strcat('DispRotwForceMomentYInputA1_data_normalMesh_smallerThickness.↙
mat'),'DispRot_F1','Indexes1')
%
clearvars -except model comsolpwd DispRot_F1 Indexes1

%%
load('forceX1_okay.mat'); range = [711,951];

model.param.set('ForceX', '0[N]'); %strcat(num2str(force),'[N]'));
model.param.set('ForceY', '0[N]');
model.param.set('ForceZ', '0[N]');
model.param.set('MomentX', '0[N*m]');
model.param.set('MomentY', '0[N*m]');
model.param.set('MomentZ', '0[N*m]');

% norms0 = sqrt(sum(force.^2, 2));
% normsAdj = sqrt(sum(forceAdj.^2, 2));
% indexMaxForce = find(normsAdj==max(normsAdj),1);
% forceAdj(indexMaxForce,:);
% thetaset1(indexMaxForce)*180/pi;

count = 0;
```

```
forRange = range(1):round((range(2)-range(1))/20,0):range(2);
for i = forRange(1:end) % minus 16 17
    count = count + 1

    model.param.set('ForceX', strcat(num2str(-forceAdj(i,1)),'[N]'));
    model.param.set('ForceY', strcat(num2str(-forceAdj(i,2)),'[N]'));
    model.param.set('ForceZ', strcat(num2str(-forceAdj(i,3)),'[N]'));
    model.param.set('MomentX', strcat(num2str(-momentAdj(i,1)),'[N*m]'));
    model.param.set('MomentY', strcat(num2str(-momentAdj(i,2)),'[N*m]'));
    model.param.set('MomentZ', strcat(num2str(-momentAdj(i,3)),'[N*m]'));

    model.sol('sol1').runAll;
    model.result('pg1').run;

    DispRot_F2(count,:) = [model.result.numerical('pev1').getReal(),model.result.↙
numerical('pev2').getReal(),model.result.numerical('pev3').getReal(), ...
        model.result.numerical('pev4').getReal(), model.result.numerical('pev5').↙
getReal(),model.result.numerical('pev6').getReal()]
    Indexes2(count) = i;
end

save(strcat('DispRotwForceMomentXInputA_data.mat'),'DispRot_F2','Indexes2')

%%%%
load('rotY_good.mat'); range = [1485,1615];

model.param.set('ForceX', '0[N]'); %strcat(num2str(force),'[N]'));
model.param.set('ForceY', '0[N]');
model.param.set('ForceZ', '0[N]');
model.param.set('MomentX', '0[N*m]');
model.param.set('MomentY', '0[N*m]');
model.param.set('MomentZ', '0[N*m]');

% norms0 = sqrt(sum(force.^2, 2));
% normsAdj = sqrt(sum(forceAdj.^2, 2));
% indexMaxForce = find(normsAdj==max(normsAdj),1);
% forceAdj(indexMaxForce,:);
% thetaset1(indexMaxForce)*180/pi;

count = 0;
forRange = range(1):round((range(2)-range(1))/15,0):range(2);
for i = forRange(1:end) %
    count = count + 1

    model.param.set('ForceX', strcat(num2str(-forceAdj(i,1)),'[N]'));
    model.param.set('ForceY', strcat(num2str(-forceAdj(i,2)),'[N]'));
    model.param.set('ForceZ', strcat(num2str(-forceAdj(i,3)),'[N]'));
    model.param.set('MomentX', strcat(num2str(-momentAdj(i,1)),'[N*m]'));
    model.param.set('MomentY', strcat(num2str(-momentAdj(i,2)),'[N*m]'));
    model.param.set('MomentZ', strcat(num2str(-momentAdj(i,3)),'[N*m]'));
```

```
    model.sol('sol1').runAll;
    model.result('pg1').run;

    DispRot_F2(count,:) = [model.result.numerical('pev1').getReal(),model.result.↙
numerical('pev2').getReal(),model.result.numerical('pev3').getReal(), ...
        model.result.numerical('pev4').getReal(), model.result.numerical('pev5').↙
getReal(),model.result.numerical('pev6').getReal()]
    Indexes2(count) = i;
end

save(strcat('DispRotwMomentYInputA_data.mat'),'DispRot_F2','Indexes2', 'forRange')

toc
```